

ҚОСТАНАЙ МЕМЛЕКЕТТІК ПЕДАГОГИКАЛЫҚ ИНСТИТУТЫ
ЖАРАТЫЛЫСТАНУ- МАТЕМАТИКА ФАКУЛЬТЕТІ
ИНФОРМАТИКА ЖӘНЕ КОМПЬЮТЕРЛІК
ТЕХНОЛОГИЯЛАР КАФЕДРАСЫ

Ерсултанова З. С.

АМАЛДЫҚ ЖҮЙЕЛЕР

Оқу құралы

Қостанай 2017

УДК 002(075.8)
ББК 32.81я73
Е 73

Автор:

Ерсултанова З. С., т.ғ.к., информатика және компьютерлік технологиялар кафедрасының доценті.

Пікір жазғандар:

Исмаилов А. О. техника ғылымдарының кандидаты, А. Байтұрсынов атындағы ҚМУ оқу үрдісін жоспарлау және ұйымдастыру басқармасының басшысы;

Еслямов С. Г. техника ғылымдарының кандидаты, доцент, ҚМПИ информатика және компьютерлік технологиялар кафедрасының меңгерушісі.

Е 73 Ерсултанова З. С.

Амалдық жүйелер: Оқу құралы.— Қостанай: ҚМПИ, 2017.— 86 б.

ISBN 978-601-7839-93-2

Ұсынылып отырған оқу құралы компьютердің жүйелік бағдарламалары-амалдық жүйелердің мәселеріне арналған. Мұнда амалдық жүйелердің негіздері, мүмкіндіктері, қызметтері және түрлері бойынша теориялық материалдар мен зертханалық жұмыстар қарастырылған.

Оқу құралы компьютерлік технологиялар саласында білім алушыларға амалдық жүйелерді меңгеруге арналып, көмекші құрал ретінде жазылды.

УДК 002(075.8)

ББК 32.81я73

Қостанай мемлекеттік педагогикалық институтының Ғылыми кеңесінің шешімімен баспаға ұсынылды.

ISBN 978-601-7839-93-2

© Ерсултанова З. С., 2017.

© ҚМПИ, 2017.

Мазмұны

Кіріспе	4
1 Бөлім. Амалдық жүйелер негіздері	
1.1 Амалдық жүйе анықтамасы	5
1.2 Амалдық жүйенің мүмкіндіктері	8
1.3 Амалдық жүйенің ресурстары	10
1.4 Амалдық жүйенің жіктелуі	12
1.5 Амалдық жүйенің жады	14
2 Бөлім. Енгізу-шығару құрылғылары	
2.1 Енгізу шығару құрылғылары	17
2.2 Амалдық жүйенің құрлары	21
2.3 Амалдық жүйеде жадты басқару	24
3 бөлім. Файлдар жүйесі	
3.1 Файлдар	29
3.2 Амалдық жүйе қауіпсіздігі	30
3.3 Ойлардың қайталанып келуі	31
3.4 Жүйелік шақырулар	33
4 бөлім. Көпдеңгейлі жүйелер	
4.1 Көпдеңгейлі жүйелер	35
4.2 Виртуалды машина	36
5 бөлім. Процестер мен ағындар	
5.1 Процестер жағдайы	39
5.2 Ағындардың қолданылуы	41
5.3 Ядродағы ағындарды іске асыру	43
5.4 Ағындар моделі	46
5.5 Процестер мен ағындар	50
5.6 Ағындар концепциясы	54
5.7 Процестің құрылуы	55
6 бөлім Амалдық жүйелер	
6.1 Windows NT/2000 амалдық жүйесі	57
6.2 UNIX амалдық жүйесі	59
6.3 LUNIX амалдық жүйесі	60
7 бөлім. Зертханалық жұмыстар	
7.1 Зертханалық сабақтар 1-9	62
7.2 Зертханалық сабақтар 10-15	74
7.3 Емтиханға дайындық сұрақтары	78
7.4 Тест тапсырмалары	80
Қолданылған әдебиеттер	86

Кіріспе

Бұл оқулықтың мақсаты ДЭЕМ-ның жүйелік ресурстары дискілерге, жедел жадыға, дисплейге, пернетақтаға қатынаудың (доступ) негізгі мүмкіндіктерінің, ресурстарға қатынаудың әр түрлі деңгейлерін пайдалана отырып, ресурстардың кейбір стандартты сипаттамаларын өзгерту мүмкіндіктерін оқып үйрену болып табылады. ЭЕМ-ң ресурстарына қатынауды әртүрлі тәсілдермен оқып үйрену: DOS функциялары деңгейінде, BIOS мүмкіндіктерін пайдалана отырып, және қажетті жерде физикалық деңгейде.

Қазіргі бөліну кемінде үш түрлі бағдарламалық қамтаманы қарастырады, ол: жүйелік; инструменталдық; қолданбалы.

Жүйелік бағдарлама – ақпаратты өңдеу жүйенің (АӨЖ) жұмысқа қабілеттігін қолдауға немесе оның тиімділігін арттыруға арналған бағдарлама, мысалы, амалдық жүйе, утилиталар, драйверлер және т.б.

Амалдық жүйе – бұл пайдаланушының талаптарын орындауға арналған компьютердің аппараттық және бағдарламалық құралдарын басаратын бағдарлама. Дамудың бірінші сатысында АЖ аппараттық құралдарды басқарумен байланысты негізгі талаптарды орындауға арналған. Бұл талаптар файлдарды басқару, бағдарламаларды орындау және пайдаланушыдан командаларды қабылдаумен шектелетін. Пайдаланушымен АЖ арасындағы әрекеттестік қолданушы интерфейсі арқылы жүзеге асырылады. Бұл интерфейс АЖ-ге қолданушының компьютерге еңгізілген инструкциясын қабылдауға және түсінуге мүмкіншілік береді.

Инструменталдық бағдарлама – бағдарламалық жүйелер және тілдер, трансляторлар және т.б.

Қолданбалы бағдарлама – пайдаланушылардың бір немесе бір топ есептерін шешетін бағдарлама, мысалы, редакторлар, баспа жүйелер, дерек қорлар басқару жүйелерінің электронды кестелері және т.б.

.Unix-жүйесінің кеңінен таралған жобасы – ол қымбат бағалы (*HP –UNIX, SUN, Solaris, DigitalUNIX*) жүйелерінің альтернативасы - арзан *OCinux.Linux*. АЖ-ң күшті жағы оның универсальдігі: жүйе қолданудың барлық диапазоның қамсыздандырады - шағын PC ден аса қуатты көп процессорлі сервер мен кластерлерге дейін. Linux АЖ-сі DOS және Windows-қа сипатталған функциялардың көбін орындайды. Бірақ бұл амалдық жүйе қуатымен және иілгіштікпен ерекшеленеді. Файлдарды басқару, бағдарламаларды басқару және пайдаланушымен әрекеттестік – ол барлық АЖ ортақ функциялар. UNIX-ң барлық нұсқаларында, соның ішінде Linux-те, тағы екі өзгешелік бар: жүйе көпесептілік және көппайдаланушылық болып табылады.

1 Бөлім. Амалдық жүйелер негіздері

1.1 Амалдық жүйе анықтамасы

Мақсаты:

Компьютерлік жүйенің деңгейлерін және құрамдарын таныстыру.

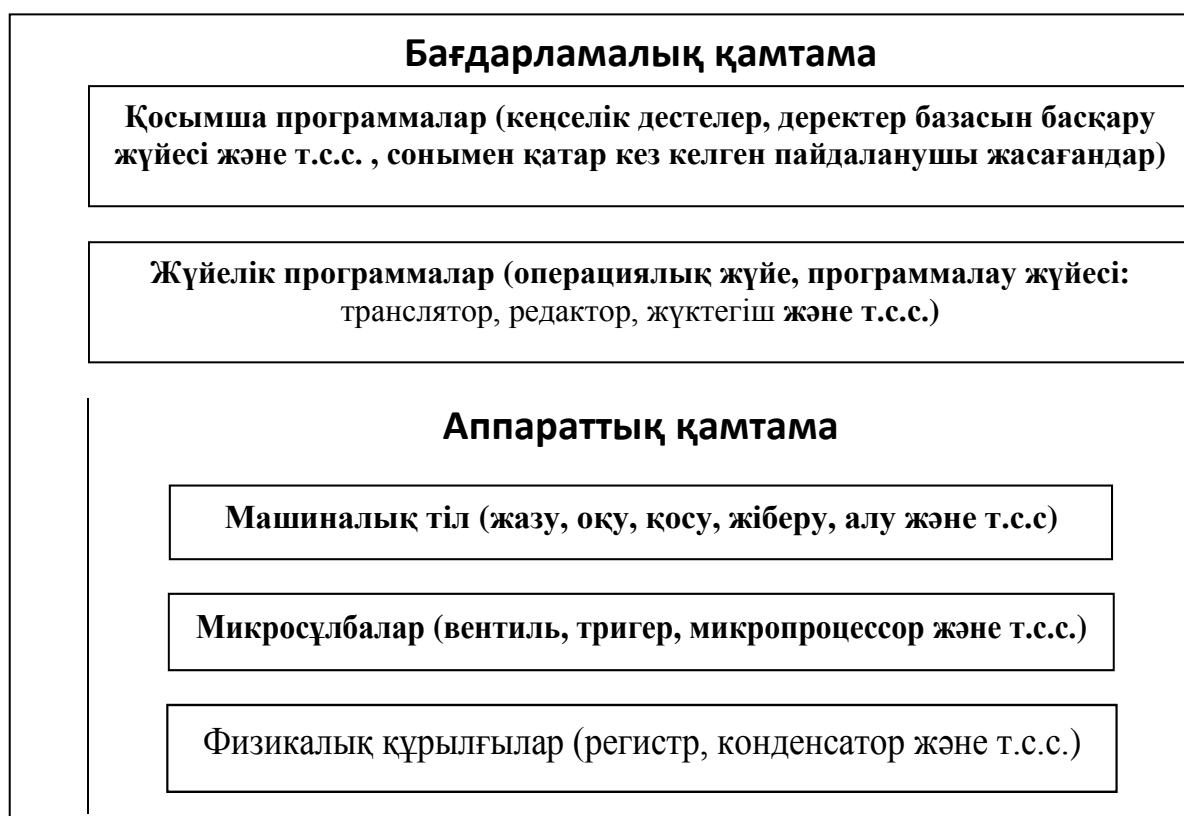
Дәріс жоспары:

1. Компьютерлік жүйенің ақпараттық қамтамасы;
2. Компьютерлік жүйенің бағдарламалық қамтамасы.

Тақырып бойынша негізгі түсінік:

Компьютерлік жүйенің аппараттық қамтамасы.

Кез-келген компьютерлік жүйе *аппараттық қамтама* және *бағдарламалық қамтама* деген екі үлкен компоненттен тұрады. Қазіргі кездегі компьютерлік жүйелердің аппараттық жабдықтамасы бір немесе бірнеше процессорлардан, оперативті жадыдан, дискіжетектен, пернеден, монитордан, принтерлерден, желілік интерфейстен және басқада құрылғылардан тұратын күрделі кешен болып табылады. Жалпы компьютерлік жүйе құрамы мен құрылымы 1-суретте көрсетілген.



1- Сурет. Жалпы компьютерлік жүйе құрамы мен құрылымы

Компьютерлік жүйенің төменгі жағында үш деңгейлі аппараттық қамтама орналасады. Оның ең төменгі деңгейін физикалық құрылғылар құрайды. Олардың жобалануы, жасалуы, орналасуы мен жұмыс жасайтындығы

электроника саласындағы инженерлік мамандарға қатысты болады. Ортаңғы деңгейде микросұлбалар орналасқан, онда физикалық құрылғылар функционалдық бірлік ретінде қарастырылады. Онда бас процессордың, оперативті жадтың, регистірлердің жұмысын жүзеге асыратын микросұлбалар орналасқан.

Ортаңғы деңгейінде бас процессордың, оперативті жадтың, регистірлердің жұмысын жүзеге асыратын микросұлбалар орналасқан. Олардың жұмыс істеу тапсырмалары белгілі командалар (бұйрықтар, нұсқаулар) тізбегінен құрылған бағдарламалар түрінде беріледі. Командалардың жиыны машиналық тілді құрайды.

Олардың жұмысын бақылау тікелей аппараттық жабдықтармен немесе бағдарламалық қосымшалармен орындалады. Бағдарламалық қосымшалардың жұмыс істеу тапсырмалары белгілі командалар (бұйрықтар, нұсқаулар) тізбегінен құрылған бағдарламалар түрінде беріледі. Командалардың жиыны машиналық тілді құрайды. Ол төменгі жақтың жоғарғы деңгейінде болады. Командаларға ақпаратты оперативті жадқа жазу, оқу, ауыстыру, өшіру сияқты амалдар мен қатар арифметикалық (қосу, алу, көбейту, бөлу және т.б.) және әртүрлі логикалық (шамаларды салыстыру және т.б.) амалдар жатады. Машиналық тілде бағдарлама жазу өте мұқияттылықты және көп уақытты талап ететін қиын жұмыс, себебі бағдарламада командалар саны мыңдаған болады. Компьютерлердің бірінші буынында осы сияқты жұмыстардың бәрін пайдаланушы өзі жасаған. Кейінгі буындарда пайдаланушыны осы қиындықтардан құтқару үшін арнаулы жасалынған жүйелік бағдарламаны амалдық жүйе деп атады.

Компьютерлік жүйенің бағдарламалық қамтамасы.

Бағдарламалық қамтаманың төменгі деңгейінде жүйелік бағдарламалар жатады. Жүйелік бағдарламалардың ішіндегі ең маңыздысы және ең күрделісі амалдық жүйе болады.

Амалдық жүйе компьютерге қосылған барлық құрылғылардың жұмысын жүргізу, басқару, бақылау және тоқтатуды дұрыс қамтамасыз етеді және тікелей аппараттық байланыстың қажет еместерін біртіндеп жояды. Мысалы, диск тіркегішіне дискідегі орнын, негізгі жадтағы адресін, оқылуы керек байт саны мен әрекет (оқу немесе жазу) бағытын жазып, оқуға немесе жазуға команда беруге болады. Іс жүзінде параметрдің көп санын беру керек, ал диск қайтаратын амал статусы айтарлықтай күрделі. Сонымен бірге енгізу-шығару құрылғыларының көбін бағдарламалау кезінде уақыттық ара қатынастары өте маңызды. Амалдық жүйе пайдаланушыға файлдарды қарауға, оқуға және жазуға қарапайым және ыңғайлы мүмкіншілік береді. Ол пайдаланушының бағдарламаласына қажетті барлық деректерді уақытысында беруге және ресурстарды тиімді пайдалануға немесе оларды қайта бөлуге мүмкіндік береді. Амалдық жүйе құрамына компьютермен бірге келген командаларды түрлендіргіш, жүйедегі

терезелер, түзеткіштер және т.б. бағдарламалар кіреді. Амалдық жүйе ядросы немесе супервизор деп аталатын режимде іске қосылатын бағдарламалар бар. Олар пайдаланушының аппарат құралдар көмегімен араласуынан қорғалған. Ал түрлендіргіштер мен түзеткіштер пайдалану режимінде іске қосылады. Көптеген жүйелерде пайдалану режимінде жұмыс істегенмен амалдық жүйелерге көмектесетін немесе арнайы міндеттер орындайтын бағдарламалар бар. Мысалы, пайдаланушыға өз рөлін ауыстыруға мүмкіндік беретін бағдарламалар жиі кездеседі. Кейбір жүйелерде, әдетте амалдық жүйелердің негізгі бөлігі (мысалы, файлдық жүйе) деп саналатынның өзі пайдаланушының кеңістігінде жұмыс істейді. Мұндай жүйелерде нақты шекараны анықтау қиын.

Жүйелік бағдарламаларға пайдаланушы бағдарламалау үшін берілетін ассемблер, транслятор, редактор, жүктегіш (загрузчик), түзегіш (отладчик) және т.б. төменгі деңгейдегі бағдарламалық камтамалар жатады. Мұндағы ассемблер машиналық тілге бірге-бір болып сәйкес келетін төменгі деңгейдегі символдық тілден, транслятор жоғарғы деңгейлі бағдарламалау тілінен машиналық тілге автоматты аударатын және пайдаланушының алғашқы кодтағы программалырының синтаксистік қателерін ұстап түзетуге мүмкіндік береді. Редактор машиналық тілге аударылған бағдарламаларды, олардың функцияларын жоғалтпай, бөлшектеуге немесе біріктруге мүмкіндік береді. Жүктегіш бөлінген немесе біріктірілген машиналық тілдегі бағдарламаларды оперативті жадтың әртүрлі немесе бір бөлігіне жүктейді. Түзегіш машина тіліндегі (объекті кодтағы) бағдарламаны жұмыс істетіп, олардың семантикалық қателерін тауып түзетуге мүмкіндік береді. Бұлардың барлығы жасаушысынан өте біліктілікті талап ететін арнаулы бағдарламалар.

Жүйелік бағдарламалардан жоғары қосымша (қолданбалы) бағдарламалар орналасқан. Көбіне олар сатылып алынады немесе пайдаланушы өзінің жекелеген мәселелерін (мәтінді өңдеу, электронды кестелер, техникалық есептеулер немесе мәліметтер базасында ақпараттарды сақтау және т.б.) шешу үшін жазады. Белгілі деңгейде амалдық жүйе екі негізгі, бірақ өзара байланыссыз міндеттерді (машина мүмкіндігін кеңейтуді және орындауды; оның ресурстарын басқаруды) жүзеге асырады.

Негізі түсінік бойынша нақты компьютерлік жүйенің құрылымымен білмейінше, онымен жұмыс істеу мүмкін емес және қазіргі дербес компьютердің кез келген қолданушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

✓ Дәріске әдістемелік нұсқау

Дәрісте берілген материалды игеру үшін компьютерлік жүйеге берілген анықтамаға көңіл бөлу керек. Негізгі ұғымдарға анықтама беріп, оларды ажырата білу керек. Дәрісте берілген негізгі мәселелер мен сұрақтарға

мән беріңіз. Берілген терминдерге анықтама беріңіз. Компьютерлік жүйенің құрылымына талдау жасаңыз.

1.2 Амалдық жүйенің мүмкіндіктері

Мақсаты:

Амалдық жүйенің екі функциясы машина мүмкіндігін кеңейту және оның ресурстарын басқару мүмкіндіктерімен таныстыру.

Дәріс жоспары:

1. Амалдық жүйе мүмкіндігі кеңейтілген машина ретінде;
2. Компьютердің ресурстарын басқару.

Тақырып бойынша негізгі түсінік:

Амалдық жүйе мүмкіндігі кеңейтілген машина ретінде.

Амалдық жүйе компьютер қосылғанда бірге жүктелетін барлық аппараттық және бағдарламалық қамтамаларының жұмысын басқарады. Ол пайдаланушы мен компьютер арасындағы байланысты қамтамасыз етеді, яғни, пайдаланушы және компьютердің электрондық құрылғылары мен бағдарламалары арасындағы келістіруші болып қызмет етеді. Ол компьютердің барлық аппараттық қамтамасының қалай жүзеге асқандығын пайдаланушыдан жасырып, оның бір есепті шешуге арналған бағдарлама орындалып жатқанда оған қажет басқа бағдарламаларды іске қосады, компьютердің ресурстарын (оперативті жақты, дискідегі орынды т.б.) бөледі.

Амалдық жүйе үзілімді (прерывание) өңдеу, таймер мен оперативтік жақты басқарумен байланысты функциялар орындайды. Сонымен қатар, басқа да төменгі деңгейлі жұмыстар жасайды.

Пайдаланушы көзқарасы бойынша амалдық жүйе аппараттық қамтамасы бар нақты компьютерге қарағанда, бағдарламалау да қарапайым және жұмыс істеуге де жеңіл мүмкіндігі кеңейтілген машина немесе виртуалды машина функциясын орындайды.

Пайдаланушының компьютерде жұмыс істеуі, шындығында оның амалдық жүйесімен диалог арқылы қатынас жасап жұмыс істеу болып табылады. Компьютерді, яғни, олардың амалдық жүйесін дүние жүзі бойынша жүздеген миллион адамдар қолданады. Қазіргі кезде компьютердің амалдық жүйесімен танысып жұмыс істеу де әдеттегі телефон шалу және теледидарды қосу, банкні және поштаны, сөздіктерді және анықтамаларды қолдана білу сияқты, қоғамда соншалықты қажетті болып келеді.

Көптеген амалдарды орындаған кезде амалдық жүйе бізді қызықтырмайтын әрекеттерді (мысалы, белгілі секторда оқитын бастиекті бос жерді тауып және сонда файлды жазуды және т.б.) жасырын орындайды. Кез - келген пайдаланушы үшін компьютермен танысудың ең ыңғайлы әдісі ол амалдық жүйе қамтамасыз ететін интерфейспен жұмыс істеу болады.

Қазіргі амалдық жүйе – бұл бірнеше жыл ішінде жүздеген және мыңдаған құрастырушылар құраған күрделі бағдарламалық кешендер. Амалдық жүйенің жазба сипаттамасы қалың томдар болады, ал толық үйрену талай жылдарға созылуы мүмкін.

Қуанышқа орай, амалдық жүйені жақсы қолдану үшін көп білудің қажеті жоқ. Ол үшін жұмыс істеудің жалпы принципін және негізгі операцияларды орындаудың әдістерін білсе болғаны. Нақты амалдық жүйемен жұмыс істеудің практикалық дағдыларын меңгермейінше, онымен жұмыс істеу мүмкін емес; жалпы принциптерін білмейінше, жүйенің тиімді қолдануды үйрене алмайсыз және қазіргі дербес компьютердің кез келген пайдаланушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

Компьютердің ресурстарын басқару

Қазіргі компьютерлер бас процессордан, оперативті жаудан, уақыт көрсеткішінен, желілік интерфейстен, дискіжетектен, пернетақадан, монитордан және басқа көптеген әртүрлі сыртқы құрылғылардан тұрады. Олардың сәулеті (командалар жүйесі, жаудардың ұйымдастырылуы, ақпараттарды енгізу және шығару және құрылғылар құрылымы) машина тілі деңгейінде ақпараттарды енгізу-шығару процестерін бағдарламалау және олармен жұмыс жасау өте ыңғайсыз. Осы бағдарламаларда ақпараттарды оқу және жазу, дискінің бастиегін жылжыту, жолдарды форматтау, сонымен қатар алғашқы рет іске қосу, тану, бастапқы жағдайға келтіру және контроллер мен жетектерді басқару командалары болады. Бұлардың ішінде оқу және жазу командалары негізгі болып табылады. Олар бірнеше параметрді талап етеді. Бұл параметрлер мынандай элементтерден : *дискідегі оқылуға тиісті блок адресі, жолдардағы секторлар саны, жазудың физикалық режимі, секторлар арасындағы аралықтарды қою*. Олар алып тасталған ақпараттар адресімен не істеу керектігін хабарлайды. Пайдаланушы сыртқы дискімен жұмыс істеу кезінде дискіжетегінің қосұлы екенін немесе қосылмағанын алдымен білуі қажет. Егер дискіжетек ағытулы болса, оны ақпараттар оқылғанша немесе жазылмағанша қосу керек.

Сыртқы дискі тозатындықтан қозғағышты ұзақ уақыт қосу күйінде қалдыруға болмайды. Пайдаланушы жүктеу кезінде орын алатын ұзақ тосу мен дискілердің тозуы (ондағы ақпараттарды жоғалтып алу ықтималдығы) арасынан таңдау жасауға мәжбүр. Осы үрдістің маңызына терең бойламай-ақ, қарапайым пайдаланушының сыртқы дискімен жұмыс істеу кезінде (қатты дискімен де жұмыс мұнан оңай емес) осындай жағдайлармен кездескісі келмейтіні түсінікті. Осылардың орнына пайдаланушыға қарапайым жоғарғы деңгейдегі абстрактылар керек. Дискілермен жұмыс істеу жағдайында дискідегі аталынған файлдар жиыны типтік абстракция болып табылады. Әр файл оқу немесе жазу үшін ашылуы, оқылған немесе жазылған болуы, ал содан

соң жабылуы мүмкін. Қозғағыштың ағымдағы жағдайы немесе толықтырылған жиілік модуляция жазу кезінде пайдалану сияқты ұғымдар пайдаланушы алдындағы абстракцияларда болмауы қажет. Осындай абстрактылармен жұмыс жасауға амалдық жүйе мүмкіндік береді. Жалпы амалдық жүйе компьютердің барлық ресурстарын басқарады. Ол компьютерлік жүйенің максималды тиімді жұмыс істеуін қаматасыз ету үшін осы ресурстарды бақылайтын процестер арасында компьютердің жадын, процессордың уақытын, сыртқы құрылғыларды және деректерді бөледі. Ресурстарды басқару олардың типтеріне байланыссыз екі үлкен есепті қамтиды:

1) Ресурстарды жоспарлау- кімге, қашан, қандай бөлінбейтін ресурстарды, ал бөлінетін ресурстардың қанша мөлшерін беру керектігін анықтайды;

2) Ресурстың жағдайын қадағалау – бөлінбейтін ресурстардың тұтыныста немесе бос болғандығы туралы, ал бөлінетін ресурстардың қандай мөлшері бөлінгендігі немесе бостығы туралы оперативті ақпаратты қолдау.

Компьютердің ресурстарын басқарудың осы жалпы есептерін шешу үшін амалдық жүйе нешәтүрлі алгоритмдерді пайдаланады. Олар компьютерлік жүйенің жалпы өнімділігін, қолдану облысын, тіпті пайдаланушы интерфейсін анықтайды.

Әр жағдайда амалдық жүйе ұсынатын абстракция, негізгі жабдықтар ұсынатынына қарағанда, анағұрлым қарапайым және қолдануға ыңғайлы. Амалдық жүйе бізге жүйелік шақырулар деп аталатын арнайы командалар көмегімен әртүрлі бағдарламаларды пайдалануға мүмкіндіктер береді.

Негізі түсінік бойынша нақты амалдық жүйенің атқаратын функцияларын меңгермейінше, онымен жұмыс істеу мүмкін емес және оны тиімді қолдануды үйрене алмайсыз.

➤ Дәріске әдістемелік сипаттама беру

Дәріспен жұмыс кезінде дәрістің негізгі түсінігіне мұқият көңіл бөліңіз. Компьютердің амалдық жүйесімен танысу қазіргі қоғамда соншалықты қажетті болып келеді. Өткізілген дәрістегі ерекшеліктерді сақтау және қызығушылықты арттыру

1.3 Амалдық жүйенің ресурстары

Мақсаты:

Амалдық жүйені пайдаланушыға интерфейс ретінде ыңғайлы, әрі тиімді жақтарын көрсету.

Дәріс жоспары:

1. Қазіргі кездегі амалдық жүйелердің ерекшелігі;
2. Бұрынғы және қазіргі кездегі компьютерлердің айырмашылықтары мен мүмкіншіліктері.

Тақырып бойынша негізгі түсінік:

Біріншіден, амалдық жүйені пайдаланушыға ыңғайлы интерфейс ретінде қарау оның компьютердегі барлық күрделі бөліктерді басқару үшін компьютер құрылымындағы механизм деген ұғымды білдіреді. Компьютерлер процессордан, жадыдан, желілік интерфейсден, принтерден және басқа көптеген құрылғылардан тұрады. Екіншіден, амалдық жүйе жұмысы оларды пайдаланатын әр түрлі бағдарламалар арасында енгізу-шығару құрылғылардың және жад пен процессорлардың бөлінуін ұйымдастыру мен бақылауды қамтамасыз ету болып табылады. Бір компьютерде үш бағдарлама жұмыс істесе және олардың бәрі бір мезгілде соңғы мәліметтер бірғана принтерде басып шығарғысы келетін жағдайды ойлап көріңіз. Мүмкін беттің басында бірінші бағдарламадан, сонан кейінгі жолдар үшінші бағдарламадан болған болар ме еді. Нәтижесінде ешкім түсінбес жағдай қалыптасар еді. Амалдық жүйе осындай жағдайларда толық тәртіп енгізеді.

Амалдық жүйе бағдарлама жұмысының барысында оның шығар мәндерін уақытша файл түрінде сақтайды. Соңынан, осы бағдарлама жұмысы біткенде, жүйе мәліметтерді принтерге жібереді. Компьютерді (немесе желіні) бірнеше пайдаланушы қолданылатын болса жадты, енгізу-шығару құрылғыларын, басқа ресурстарды басқару және оларды қорғау қажеттегі күрт артады. Өйткені пайдаланушылар оларға кез-келген ретпен шығуы мүмкін. Пайдаланушылар арасында тек жабдықтарды ғана емес, ақпараттарды (файлдар, деректер базасы және с.с.) бөлуге тура келеді. Осы көзқарас тұрғысында амалдық жүйе міндеті кім және қандай ресурс пайдаланатындығын, ресурстарға сұраныстарды өңдеу, жүктеу коэффициентін есептеуді және әртүрлі бағдарламалармен пайдаланушылардың қарама-қайшылықтар мәселелерін қадағалап отыру болып табылады.

Ресурстарды басқару екі тәсілмен жүзеге асады: уақыт бойынша және кеңістік бойынша. Ресурс уақыт бойынша, бөлінген жағдайда, әр түрлі пайдаланушылар оларды кезекпен пайдалануға мүмкіндік алады, одан соң басқасы т.с.с. Мысалы, бірнеше бағдарламалар орталық процессорға шығады. Бұл жағдайда амалдық жүйе алдымен процессорға бір бағдарламамен жұмыс істеуге мүмкіндік жасайды, соңынан ол жеткілікті уақыт істеген соң, басқа бағдарламаға, одан соң келесіге, ең аяғында қайтадан біріншіге береді. Ресурс уақыт бойынша қанша рет пайдаланылады, келесі кім болады және оған ресурс қанша уақытқа беріледі. Осының барлығына амалдық жүйе міндетті.

Негізі түсінік бойынша нақты амалдық жүйемен жұмыс істеудің практикалық дағдыларын меңгермейінше, онымен жұмыс істеу мүмкін емес; жалпы принциптерін білмейінше, жүйенің тиімді қолдануды үйрене алмайсыз және қазіргі дербес компьютердің кез-келген қолданушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

➤ Дәріске әдістемелік сипаттама беру

Дәріспен жұмыс кезінде дәрістің негізгі түсінігіне мұқият көңіл бөліңіз. Дербес компьютердің амалдық жүйесімен танысуда сөздіктерді және анықтамаларды қолдана білу сияқты, қазіргі қоғамда соншалықты қажетті болып келеді. Өткізілген дәрістегі ерекшеліктерді сақтау және қызығушылықты арттыру.

1.4 Амалдық жүйенің жіктелуі

Мақсаты:

Амалдық жүйенің әртүрлі жіктелуінің мүмкіндіктерімен таныстыру.

Дәріс жоспары:

1. Амалдық жүйенің бір есепті және көп есептілігі.

2. Амалдық жүйенің бір пайдаланушылық және көп пайдаланушылық болып жіктелулері.

Тақырып бойынша негізгі түсінік:

Амалдық жүйелер майнфреймдік, серверлік, көпроцессорлық, нақты уақтылық, дербес компьютерлік, ендірілген деп жіктеледі.

Мэйнфреймдердің енгізу-шығару мүмкіндіктері өте жоғары, олар мыңдаған дискілермен және көлемдері терабайт болатын деректермен жұмыс істейді. Мэйнфреймдік амалдық жүйе бір мезгілде көп тапсырманы (негізінен енгізу-шығару амалдары) орындауға бағытталады және ол мынадай үш *дестелік өндеу, транзакцияны өндеу (топтық операциялар)*, және *уақытты бөлу* режимде жұмыс істейді.

Серверлік амалдық жүйелер бір мезгілде көп пайдаланушыларға қызмет көрсетеді және оларға өзара бағдарламалық және аппараттық ресурстарды бөлуге мүмкіндік береді.

Көпроцессорлық жүйеде екі немесе бірнеше орталық процессорлар бір ортақ оперативті жадқа толық қолжеткізуді бөледі, яғни, бүкіл жадты адрестеуді бөледі.

Нақты уақытты амалдық жүйелерде неізгі параметр уақыт болады. Олар қатал уақытты жүйе немесе икемді уақытты жүйе болып екіге бөлінеді. Біріншісінде орындалатын амал немесе көрсетілетін қызмет тек дәл анықталған уақытта орындалуы міндетті (оларды уақыттан кешіктіріп орындауға мүлде болмайды), ал екіншісі орындау уақытын жыжытуға болады.

Дербес компьютерлік амалдық жүйелер бірнеше параметрлер бойынша ерекшеленеді. Бір есепті амалдық жүйе әрбір уақытта адамға компьютерде бір ғана есепті шешуге, яғни, тек бір ғана іспен айналысуға мүмкіндік береді. Бір есепті жүйелер қарапайым, шағын, ең аз қуатты компьютерлерде жұмыс істей алады, бірақ жұмыс істеу қолайлылығы жағынан көп есептіден ұтылады, сондықтан оларға жылдам жол беріп, ығыстырылып келеді. Көп есепті амалдық жүйе бір мезгілде бірнеше бағдарламаларды іске қосуға мүмкіндік береді. Бұл бағдарламалар бір-біріне кедергі келтірмей, қатар (параллель) жұмыс істейтін болады.

Бір пайдаланушылық амалдық жүйе компьютерде тек бір ғана адамға жұмыс істеуге мүмкіндік береді. Көп пайдаланушылық жүйеде әрбір пайдаланушы барлық ақпараттармен және құпиясөз енгізу арқылы тек өзіне қатысты жеке ақпараттармен байланыса алады.

Кейбір көп пайдаланушылық жүйелер, мысалы UNIX, бір мезгілде бірнеше пайдаланушыларға бір компьютерде жұмыс істеуге мүмкіндік береді. Бұл жағдайда әрбір пайдаланушының жұмыс орнында терминал орнатылуы қажет. Терминал қызметтік не арнайы бағытталған құрылғылар (пернетақтасы бар бейне монитор), не дербес компьютерде жүктелген арнайы бағдарлама орындауы мүмкін. Пайдаланушылар өз терминалдарында бір-бірін байқамай және кедергі келтірмей, бір мезгілде бір компьютермен ғана жұмыс істей береді.

Терминал компьютерден қашықта (бірнеше метрде немесе бірнеше мың километрде) орналасуы мүмкін. Егер мекеме компьютерінде көп пайдаланушылық амалдық жүйе және бірнеше модемдер орнатылған болса, онда әрбір қызметкер компьютер мен модем арқылы байланыса алады және өзінің дербес (жұмыс орындық немесе тасымалданатын) компьютерін терминал ретінде қолдана отырып, ол өзінің жұмыс орнында отырғандай жұмыс істей алады. Терминал негізгі компьютермен және дүниежүзілік компьютерлік желі Internet арқылы байланыса алады.

1990 жылдарға дейін көп адамның амалдық жүйемен қарым-қатынасы командалық режимде болған: пайдаланушы жүйенің шақыруына жауап ретінде команданы теретін, жүйе оны орындайтын. Мұндай командалық режим көптеген командалардың пішімдерін білуді және ұзын командалық қатарды қатесіз енгізуді талап ететін.

Соңғы жылдары графикалық көп терезелі интерфейс жиі қолданылады, мұнда қажетті әрекеттер және объектілердің сипаттамасы мәтін түрінде енгізілмейді, мәзірден, файлдар тізімінен немесе маус арқылы таңдалады. Бұл көп терезелі интерфейс бір мезгілде бірнеше бағдарламалармен жұмыс істегенде ерекше қолайлы: олардың әрқайсысымен адам жеке терезеде жұмыс істейді.

Бұдан бірнеше жыл бұрын Microsoft компаниясының қарапайым бір есепті, бір пайдаланушы командалық интерфейсі бар MS-DOS амалдық жүйесі ең танымал болған. Бұл жүйе дербес компьютерде ең қарапайым ресурстармен жұмыс істеуге қабілетті болатын.

Бүгінгі дербес компьютерлердің қуаты сонша күшті болғандықтан, оған көп терезелі графикалық интерфейсі бар қуатты, көп есепті амалдық жүйені орнату тиімді. Олардың ішінде ең көп тарағаны Windows.

Сонымен қатар, амалдық жүйелердің кішірейтілген версиялары қалталы компьютер немесе қалтаға сыйатын шағын (телефондық және жазба кітапшасы және блокнот) қызмет орындайтын кішкентай компьютерлерде ендіріліп орнатылады.

➤ Қорытынды шығару.

Негізі түсінік бойынша нақты амалдық жүйемен жұмыс істеудің практикалық дағдыларын меңгермейінше, онымен жұмыс істеу мүмкін емес; жалпы принциптерін білмейінше, жүйенің тиімді қолдануды үйрене алмайсыз және қазіргі дербес компьютердің кез келген пайдаланушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

➤ Дәріске әдістемелік сипаттама беру

Дәріспен жұмыс кезінде дәрістің негізгі түсінігіне мұқият көңіл бөліңіз. Дербес компьютердің амалдық жүйесімен танысуда сөздіктерді және анықтамаларды қолдана білу сияқты, қазіргі қоғамда соншалықты қажетті болып келеді. Өткізілген дәрістегі ерекшеліктерді сақтау және қызығушылықты арттыру

1.5 Амалдық жүйенің жады

Мақсаты:

Амалдық жүйенің жадтары және олардың құрамы мен құрылымын таныстыру.

Дәріс жоспары:

1. Амалдық жүйе жадтарының иерархиясы;
2. Амалдық жүйе жадтарының ерекшеліктері.

Тақырып бойынша негізгі түсінік:

Амалдық жүйе жадтарының иерархиясы.

Амалдық жүйенің негізгі аппараттық қамтамасының бірі болып жад жүйесі есептеледі. Жад жүйесі төменгі кестеде көрсетілгендей иерархиялық деңгейден тұрады. (1-кесте)

1-кесте. Жад түрлері

№	Жад түрлері	Жадтың орта көлемі	Жадқа қолжеткізудің орта уақыты
1.	Регистрлер	1 Кбайттан кіші	1 нс
2.	Кэш-жады	1 Мбайт	2 нс
3.	Негізгі жад	64 Мбайт -1 Гбайт	10 нс
4.	Қатқыл диск	20 - 100 Гбайт	10 мс
5.	Сыртқы диск	1-50 Гбайт	100 мс

Бұл кестенің бірінші жолында бас процессордың ішкі регистрлерінің сипаты келтірілген. Олар процессорлар сияқты өте жылдам жұмыс

істейді. Ішкі регистрлер n разрядты процессорда $n \times n$ бит ақпаратты сақтайды. Қазір 64-разрядты процессорлар бар, сондықтан, оның ішкі регистрінде 64×64 бит ақпаратты сақтауға болады. Бағдарламалар аппараттық құралдардың қатысуынсыз регистрлерді басқара алады.

Кестенің екінші жолында кэш-жадының сипаты жазылған. Бұл жад аппараттық құралдарды басқарады. Оперативтік жад, әдетте, әрқайсысы 64 байтты бірнеше кэш-жолдарға бөлінген. Оның нөлінші жолы 0-ден 63-ке дейін, бірінші жолы 64-тен 127-ге дейін, екінші жолы 128-ден 191-ге дейін және қалғандыры осы сияқты адрестелген. Ең көп қолданылатын кэш-жолдар процессордың ішінде орналасқан жоғарғы жылдамдықты кэш-жадында сақталады. Кейбір компьютерде екі немесе үш деңгейлі кэш-жадылар бар. Олардың әрбір келесісі алдыңғысынан гөрі үлкен, бірақ баяу жұмыс істейді.

Үшінші жолда негізгі жад сипатталған. Ол негізінен оперативті жадтан және тұрақты жадтау құрылғысынан (ТЖҚ) тұрады. Оперативті жадты кейде еркін қолжеткізулі жад RAM (Random Access Memory) деп атайды. Оперативті жадқа қатқыл немесе сырқы дискіден осы кезде орындалатын бағдарлама көшіріледі. Оперативті жадтың бір түрі ол бейне жад VRAM. Онда цифрлармен кодталған бейнелелер сақталады. Оның ішіндегісіне бір мезгілде екі құрылғы процессор мен монитордың қолжеткізуі жүзеге асырылған. Сондықтан экрандағы бейне жадтағы бейненің бағдарлама арқылы өзгеруімен бір мезгілде өзгереді.

ТЖҚ энергияға байланысты жад және оған жазылатын ақпарат ешқашан өзгермейтін болады, яғни, ол тек қана оқуға арналған. Бұл жадқа ақпарат арнаулы әдіспен компьютер жасалғанда жазылады. Оның ішінде мониторды, пернетақтаны және принтерді басқаратын бағдарлама, сонымен қатар, компьютерді жіберетін, тоқтатын және құрылғыларды тестілейтін бағдарламалар жазылады. Қайта бағдарламалауға болатын тұрақты жад энергияға байланысты және оның ішіндегісін бірнеше рет қайта жазуға болатын физикалық құрылғы. Ең алдымен бұл жадқа процессордың өзін басқаратын бағдарлама жазылады. Оның ең маңызды микросұлбасын базалық енгізу/шығару жүйесі BIOS (Basic Input/Output System) деп атайды. Ол компьютерді жібергеннен кейін құрылғыларды автоматты тестілеуге, оперативті жадқа амалдық жүйені жүктеуге арналған бағдарламалар жиынынан тұрады. BIOS рөлі екі жақты: бір жағынан ол аппараттық қамтаманың бөлігі, ал екінші жағынан ол бағдарламалық қамтама. Тұрақты жадтың тағы бір түрі жылдамдығы төмен және энергияны аз (батареядан) тұтынатын CMOS RAM. Ол компьютердің конфигурациясы, жабдықтар құрамы мен жұмыс істеу режимі туралы ақпаратты және күн мен уақытты сақтауға арналған. Оның ішіндегісін BIOS құрамына кіретін арнаулы Setup бағдарламасымен өзгертуге болады.

Амалдық жүйе жадтарының ерекшеліктері

Оперативті жадтың жалғыз қатар дәнекерлі жад модулы SIMM (Single In-line Memory Module) және қос қатар дәнекерлі жад модулы DIMM (Dual In-line Memory Module) бар.

Алдымен 8-битті 32 дәнекерлі және 32-битті 72 дәнекерлі SIMM-дер шығарылды. Кейін 64-битті 168 дәнекерлі DIMM-дер пайда бола бастады. Қазыр Rambus жады үшін RIMM модулы жасалды. Онда модульға қойылатын талап өте дәлдікпен жазылып күшейтілді. Модуль үшін жеке технология қарастырылды.

Қатқыл дискі ақпаратты тұрақты сақтау үшін арналған және бір немесе бірнеше магниттелген табақшалардан тұрады. Табақшаның магниттелген беті жолдар деп аталатын жіңішке концентрлі сақиналық зоналарға, ал әрбір жол секторларға бөлінеді. Секторды шартты түрде деректер облысына және қызмет ақпараты облысына бөледі. Қызмет ақпаратын дискі жасаушы бір рет ғана жазады, ол кейін өзгермейтін болады. Қызмет облысы контроллер ақпаратты оқыған және жазған кезде танитын сектордың бірегей адресін қамтиды. Деректер облысы дискіге жазылатын пайдалы ақпаратты қамтиды. Бұл облыс дискіні пайдаланған кезде көп рет өзгертіледі. Деректер облысының көлемі ақпараттың көлемінен біраз үлкен болады. Себебі, онда ақпаратты қосымша тексеру және оның қатесін түзету үшін орын керек. Сектордың деректер облысы тек бүтіндей өзгертіледі. Бірнеше табақшалардағы диаметрлері бірдей сақиналарды цилиндр деп атайды. Дискілік жүйенің тиімділігін арттыру үшін деректерді бір цилиндрде орналастырған жөн.

Негізгі сырқы дискілерге магнитті оптикалық дискілер (МО-дискілер) және оптикалық дискілер жатады. МО-дискілерге ақпаратты жазу лазер мен магниттік бастиектің өзара әрекеттесуі арқылы жүзеге асырылады. Бұл дискілер магниттік және температуралық өрістердің тербелістеріне шыдамды болады. Оптикалық дискілердің жұмыс істеу принципінде лазерлік сауле қолдану жатыр. Бұл дискілерде жазылған цифрлық деректерді олардың беттеріне шағылған лазерлік сауле арқылы оқиды. Жазылған цифрлік деректердің тығыздығы магниттік дискілердігінен әлдеқайда жоғары. Оптикалық дискіге CD-дискі (Compact Disk) мен DVD-дискі (Digital Video Disk) жатады. CD-дискілер мен жұмыс істейтін тек оқуға арналған құрылғыны CD-ROM (Compact Disk Read Only Memory) деп, ал оқу мен жазуға арналғанды CD-R (Compact Disk Recordable) деп атайды. Стандартты CD-дискілердің көлемі 650-700 Мбайт. DVD-дискі қайта жазуды қолдайды, оның көлемі 4,7 – 9,4 Гбайт.

Негізі түсінік бойынша нақты амалдық жүйенің жадтарының түрлерін және олардың жұмыс істеу принциптерін білмейінше, онымен жұмыс істеу мүмкін емес және қазіргі дербес компьютердің кез келген пайдаланушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

2 Бөлім. Енгізу- шығару құрылғылары

2.1 Енгізу шығару құрылғылары

Мақсаты:

Енгізу/шығару құрылғылары және олардың түрлері, құрамы және қасиеттерімен таныстыру.

Дәріс жоспары:

1. Енгізу/шығару құрылғыларының категориялары;
2. Енгізу/шығару құрылғыларының жылдамдықтары;
3. Енгізу/шығару құрылғыларының бөліктері.

Тақырып бойынша негізгі түсінік:

Енгізу/шығару құрылғыларының категориялары.

Компьютерлік жүйеде бірнеше енгізу/шығару құрылғылары бар.

Оларды үш категорияға бөлуге болады: *блоктық құрылғы, символдық құрылғы, ерекше құрылғылар.*

Блоктық құрылғылар ақпаратты бекітілген мөлшерлі және өзінің адресі блок түрінде сақтайды. Әдетте блок мөлшері 512 байттан 32 768 байтқа шейін болады. Әрбір блок басқа блоктарға байланыссыз оқылады. Көп тараған блоктық құрылғыға дискілер жатады.

Символдық құрылғылар символдар ағынын ешбір блоксыз қабылдай және жібере алады. Олар адрестелінбейді және іздеу амалын орындамайды. Оларға принтерлер, интерфейстік желілік қарталар, тышқандар мен дискілерден басқа құрылғылар жатады.

Ерекше құрылғылардың жұмыс істеу принциптері блоктық және символдық құрылғылардан өзгеше болады. Оларға қатаң анықталған уақытта үзілім жасайтын таймер, жадты бейнелейтін экран және тағы басқалары жатады.

Енгізу/шығару құрылғыларының жылдамдықтары

Өртүрлі енгізу/шығару құрылғыларының жылдамдықтары төменгі кестеде көрсетілген: (2-кесте)

2-кесте. Құрылғы жылдамдықтары

№	Құрылғы	Жылдамдық
1.	Пернетақта	10 байт/сек
2.	Маус	100 байт/сек
3.	Модем 56 К	7 Кбайт/сек
4.	Телефон	8 Кбайт/сек
5.	ISDN	16 Кбайт/сек
6.	Лазерлік принтер	100 Кбайт/сек
7.	Сканер	400 Кбайт/сек
8.	Классикалық Ethernet	1,25 Мбайт/сек
9.	USB құры	1,5 Мбайт/сек

10.	Цифрлы бейне камера	4 Мбайт/сек
11.	IDE-диск	5 Мбайт/сек
12.	40x CD-ROM	6 Мбайт/сек
13.	Жылдам Ethernet	12,5 Мбайт/сек
14.	ISA құры	16,7 Мбайт/сек
15.	IDE (ATA-2)	16,7 Мбайт/сек
16.	FireWire (IEEE 1394)	50 Мбайт/сек
17.	XGA-монитор	60 Мбайт/сек
18.	SONET OC-12 желісі	78 Мбайт/сек
19.	SCSI OC-12 дискі	80 Мбайт/сек
20.	Гигабитті Ethernet желісі	125 Мбайт/сек
21.	Ultrium таспасы	320 Мбайт/сек
22.	PCI құры	528 Мбайт/сек
23.	Біріккен Sun Gigaplane XВ тақташасы	20 Гбайт/сек

Енгізу/шығару құрылғыларының бөліктері

Енгізу/шығару құрылғылары электрондық және механикалық бөліктен тұрады.және құрылғының өзі электрондық бөлік *контроллер* немесе *адаптер (бейімдегіш)* деп аталады. Ол алынып салынатын микросұлбалар тақташасы түрінде жасалады және амалдық жүйенің командасын (мысалы, деректерді оқу нұсқауын) қабылдап орындау арқылы құрылғыларды физикалық басқара алады. Кейбір контроллер бірнеше бірдей құрылғыларды басқара алады. Бірақ көп жағдайда құрылғыларды нақты басқару өте қиын және жоғарғы деңгейдегі айқындауды талап етеді. Механикалық бөлік құрылғының өзі болады. Енгізу шығару құрылғыларының мынадай түрлері бар: қатты диск, магнитті оптикалық (МО) диск, иілгіш диск, CD/DVD-диск, монитор, пернетақта, принтер, сканер және тағы басқалар. Құрылғылар интерфейсі қарапайым болады, себебі оның мүмкіншіліктері шектелген және оларды бір стандартқа келтіру қажет.

Қатты дискі контроллері оқу командасын қабылдағанда оған сектордың сызықты нөмірі ғана беріледі. Оны орындау үшін контроллер мыналарды орындау қажет:

1) Сызықты сектор нөмірін цилиндр, сектор және бастиек нөміріне түрлендіру (мұндай түрлендірудің қиындығы сыртқы цилиндрлердің ішкі цилиндрлерге қарағанда көп секторы болуына және бүлінген секторлардың нөмірлері басқа секторларда бейнеленуіне байланысты);

2) Бастиектің қай цилиндрді көрсетіп тұрғандығын анықтау және оны қажет секторға жылжыту үшін импульстер тізбегін беру;

3) Сектордағы ақпарат орналасқан биттерді олардың келу ретіне байланысты оқу мен сақтау, тақырыпты жою және бақылау қосындысын есептеу;

4) Қабылданған барлық биттерді сөздерге жинау жән оларды негізгі жадта сақтау.

Бұл жұмыстардың барлығын атқару үшін контроллер осындай жұмысқа бағдарламаланған кіші мөлшерлі қарапайым ендірілген компьютерледі қамтиды. Сондықтан контроллер құрылғыларды басқару міндетін өзінің амалдық жүйеге арналған интерфейсі арқылы жүзеге асырады. Контроллерге арнаулы бағдарламаылық қамтама қажет. Ол әртүрлі контроллер үшін жеке жазылады. Контроллермен қатынас жасайтын бағдарлама құрылғы драйвері деп аталады. Ол құрылғыға нұсқау береді және оның жауабын күтеді. Әрбір контроллер жасаушылары белгілі бір амалдық жүйелер қолдайтын драйверді қоса жасау керек. Егер сіз драйверді пайдаланғыңыз келесе, онда ол амалдық жүйенің ядролық режимінде жұмыс істейтіндей қылып жүйе құрамына орнату керек. Драйверлер амалдық жүйенің ядросына үш түрлі жолмен орнатылады:

1) Ядроны қажет драйверлерді қосып, оны қайта жинақтағаннан кейін амалдық жүйені қайта жүктеу қажет (осылай Unix жұмыс істейді);

2) Амалдық жүйе құрамына кіретін файлда драйвер қажеттігі туралы жазба жасау керек те амалдық жүйені қайта жүкте қажет. Амалдық жүйе бастапқы жүктелу кезінде қажет драйверлерді тауып, оларды өзі жүктейді (осылай Windows жұмыс істейді);

3) Амалдық жүйе өзінің жұмысын үзбей және қайта жүктелуді талап етпей жаңа драйверлерді қабылдап, оларды жедел орнатады. Бұл драйверлерді динамалық жүктелетін драйверлер деп атайды.

Әрбір контроллермен байланыс орнату үшін бірнеше регистрлер бар. Олар жадтағы адресті, дискідегі адресті, сектор нөмірін және амал бағытын (оқу немесе жазу) анықтау үшін қолданады. Контроллерді белсендіру үшін драйвер амалдық жүйеден команда алады, сонан кейін оны құрылғы регистріне жазатындай шамаға аударады.

Кейбір компьютерлерде құрылғы регистрлері амалдық жүйенің адрестік кеңістігінде бейнеленеді. Сондықтан оларды әдеттегі жад сөздері сияқты оқуға немесе жазуға болады. Мұндай компьютерлерде арнайы енгізу/шығару командаларының қажеті жоқ, ал осы адрестерді бағдарламалар қатынас жасай алмайтын жерлерге (мысалы, базалық регистрге) жазып, пайдаланушылардың бағдарламаларын аппараттық жабдықтардан шектеуге болады. Бұлар арнаулы енгізу/шығару командалар болуын қаламайды, бірақ адрестік кеңістікті алап етеді.

Басқа компьютерлерде құрылғылар регистрі арнаулы нгізу/шығару порттарында орналасады және оларда ядро режимінде енгізу/шығару командасына қолжеткізу бар. Мұндай компьютерлер драйверлерге ақпаратты регистрлерден оқуға және оларға жазуға мүмкіндік береді. Бұлар адрестік кеңістікті керек қылмайды, бірақ арнаулы енгізу/шығару командалары болуын талап етеді.

Деректерді енгізі/шығару үш әдісіпен жүзеге асырылады:

1) *Белсенділікті күту немесе дайындықты күту әдісі.* Пайдаланушының бағдарламасы жүйелік сұратым береді, жүйе ядросы сәйкес драйверлерінің процедураларына аударады. Сонан кейін, драйвер енгізі/шығару процесін бастайды. Осы уақытта драйвер қажет құрылғының дайындығын сұрайтын өте қысқа циклды бағдарламаны орындайды. Енгізі/шығару амалы орындалып біткеннен кейін драйвер қажет жерден/жерге деректерді оқиды/орналастырады да өзінің алғашқы жағдайына оралады. Соңында, амалдық жүйе басқаруды сұратым жасаған пайдаланушы программасына қайтарады.

2) *Бағдарламалық үзілімді қолдану әдісі.* Драйвер құрылғыны жібереді және енгізі/шығару біткеннен кейін үзілімді беруді сұрайды. Осыдан кейін драйвер деректерді қайтарады, амалдық жүйе шақырған бағдарламаны бұғаттайды және, егер керек болса, басқа тапсырмаларды орындайды. Контроллер деректерді беру аяқталғандығын байқағанда, амалдың аяқталғандығы туралы хабар беру үшін үзілімді тудырады.

3) *DMA тура қатынау жадының арнаулы контроллерін қолдану.* Бұл контроллер оперативті жад пен кейбір контроллер арасындағы ағынды орталық процессорды тұрақты араластырмай басқарады. Процессор DMA контроллерін шақырып, оған қанша байтты жіберу керектігін анықтап, құрылғы мен жадтың хабарлап, деректерді жіберу бағытын (енгізу/шығару) көрсетіп алып, ары қарай оның өзі әрекет жасауына мүмкіндік береді.

Кейбір үзілімдер басқа үзілімді өңдеп жатқанда пайда болуы мүмкін. Орталық процессордың бұларды өңдеуге тиым салып, оған рұқсатты кейін беретін мүмкіншілігі бар. Үзілімдерге тиым салынып тұрғанда жұмыстарын аяқтаған барлық құрылғылар өзінің сигналдарын процессорға жібере береді, бірақ одан процессордың жұмысы үзілімдер рұқсат алмайынша үзілмейді. Егер үзілімдерге тиым салынып тұрғанда бірнеше құрылғылар жұмыстарын бірден аяқтаған болса, онда үзілімдер контроллері олардың қайсысы бірінші шешу үшін олардың артықшылығын қолданады. Жоғары артықшылығы бар құрылғыға алдымен қызмет көрсетіледі.

Негізі түсінік бойынша нақты компьютерлік жүйенің енгізу/шығару құрылғыларының түрлерін және олардың жұмыс ітеу принциптерін білмейінше, олармен жұмыс істеу мүмкін емес және қазіргі дербес компьютердің кез келген пайдаланушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

✓ Дәріске әдістемелік нұсқау

Дәрісте берілген материалды игеру үшін компьютерлік жүйенің енгізу/шығару құрылғыларының түрлері, олардың ерекшеліктеріне көңіл бөлу керек. Негізгі ұғымдарға анықтама беріп, оларды ажырата білу керек. Дәрісте берілген негізгі мәселелер мен сұрақтарға мән беріңіз.

Берілген терминдерге анықтама беріңіз. Компьютерлік жүйенің енгізу/шығару құрылғыларына талдау жасаңыз.

2.2 Амалдық жүйенің құрлары

Мақсаты:

Амалдық жүйенің құрларының түрлері, функциялары және басқа қасиеттерімен таныстыру.

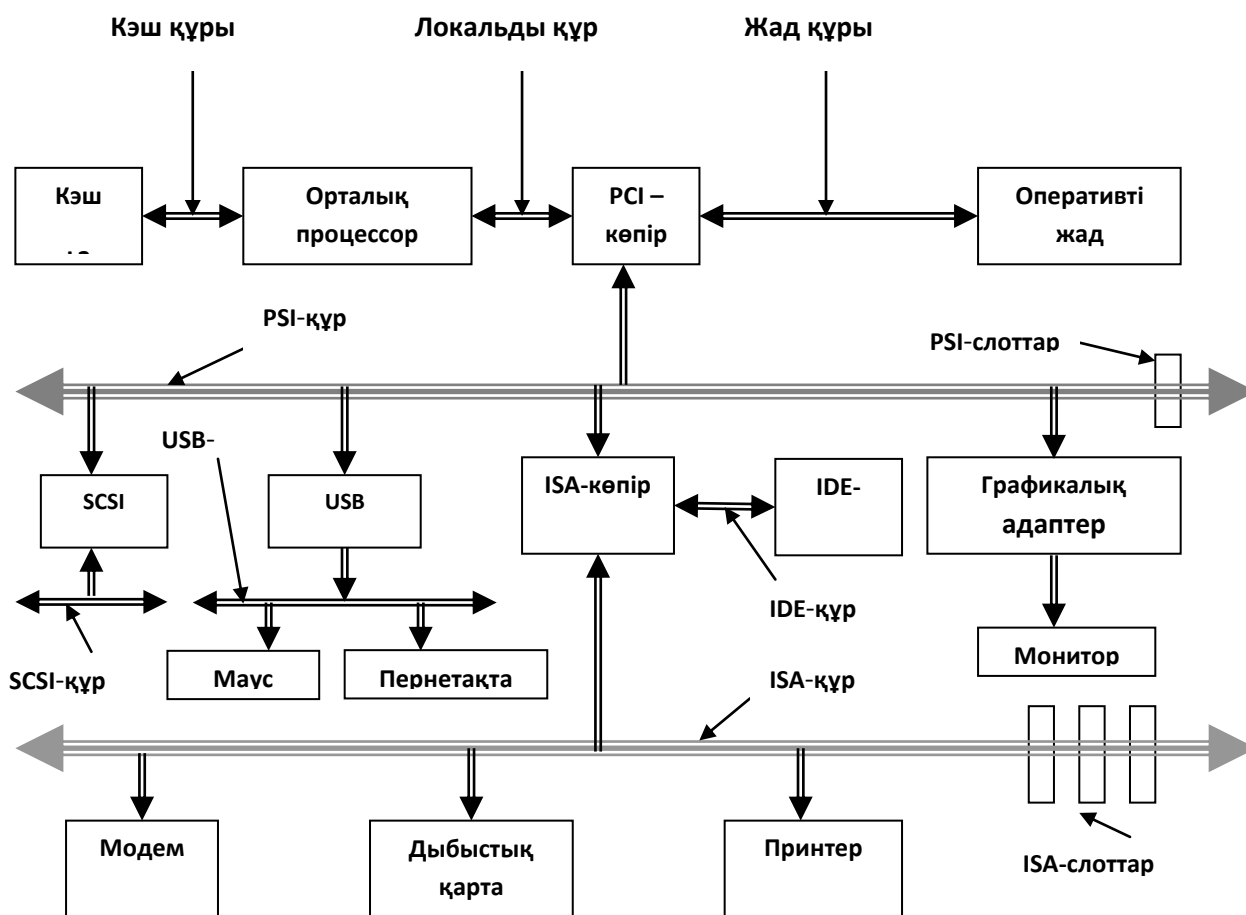
Дәріс жоспары:

4. Құрлардың түрлері мен сипаты;
5. Енгізу/шығару құрылғыларымен жұмыс істеуі.

Тақырып бойынша негізгі түсінік:

Құрлардың түрлері мен сипаты.

Pentium компьютерінің үлкен жүйесінің құрылымында сегіз құр бар: кэш құры, локалды құр, жад құры, PCI құр, SCSI құр, USB құр, IDE құр, ISA құр. Амалдық жүйеде компьютерді басқару үшін барлық аталған құрлар туралы ақпарат болуы қажет. Төменде Pentium компьютерінің үлкен жүйесінің архитектурасы көрсетілген.



Pentium үлкен жүйесінің құрылымы

Құрлардың әрқайсы өзінің функциялары және жылдамдықтары бойынша бір-бірінен айырылады. Көрсетілген құрлардың ішінде екі негізгі құр бар. Олар өндірістік стандартты архитектура – Industry Standard Architectura (ISA) және шеттегі құрылғылар интерфейсі – Peripheral Component Interconnect (PCI). ISA құр ең алғаш IBM PC/AT компьютерінде пайда болды, оның жылдамдығы 16,67 Мбайт/сек. PCI құр ISA құрдың ізбасары ретінде жасалды, оның жылдамдығы 528 Мбайт/сек. Оған жылдам жұмыс істейтін құрылғылар қосылады. Жүйеде үш арнаулы SCSI, USB, IDE құрлар бар. Мұндағы SCSI – шағын компьютерлердің жүйелік құры, ол жіберу мүмкіншілікті молынан талап ететін жылдам дискілерге, сканерге және басқа құрылғыларға арналған жоғарғы өнімді арнаулы құр, оның жылдамдығы 160 Мбайт/сек. USB - әмбебап тізбекті құр, оның жылдамдығы 1,5 Мбайт/сек., ол компьютерге барлық баяу жұмыс істейтін енгізу/шығару құрылғыларын (пернетақта, маус, принтер және т.б.) қосу үшін арналған. IDE құр жүйеге сыртқы дискі және CD/DVD дискіден/ге оқитын/жазатын құрылғыларды қосу үшін арналған. Сонымен қатар, Apple компаниясы жасаған тағы FireWire немесе IEEE 1394 құрды қодануға болады. FireWire құр USB құры сияқты тізбекті болады, бірақ ол жылдамдығы 50 Мбайт/сек. жететін деректердің дестемен жіберілетіндігін қолдайды. Оған цифрлық бейнекамера және басқа мультимедиялық құрылғыларды қосуға болады.

Компьютердің жүктелуі.

Компьютердің осы конфигурациясында орталық процессор локалды құр арқылы PCI-көпірге деректер жібереді, ол ары қарай жад құры арқылы оперативті жадпен қатынас жасайды. Бұл компьютерде кэш жадының екі деңгейі бар: L1 – ендірілген кіші кэш, L2 – сыртқы үлкен кэш. Орталық процессор соңғысымен арнаулы кэш құры арқылы қатынас жасайды.

Амалдық жүйе құрылғыларды өзі тауып, оларды баптай алуы қажет. Ол үшін Intel мен Microsoft компаниялары қос та жұмыс істе - plug and play атты дербес компьютерінің жүйесін жасап шықты. Оны жасаудың алдында әрбір енгізу/шығару тақташасында енгізу/шығару регистрлерінің бекітілген адрестері және үзілімнің сұратым деңгейі болды. Егер әртүрлі құрылғы бір үзілімді пайдаланса, онда олар бір-бірімен қарама қайшылыққа келеді. Оларды шешу үшін арнаулы тұйықтағыш – jumper қолданды. Кейін бұл проблема plug and play стандартында автоматты шешілді, ол үшін алдымен жүйедегі барлық енгізу/шығару құрылғылары туралы ақпарат жиналды, бір ортадан үзілім деңгейлері мен енгізу/шығару адрестері тағайындалды, сонан кейін осыны әрбір тақташаға жеткізді. Әрбір компьютердің аналық тақташасында BIOS деген бағдарлама болады. Онда төменгі деңгейдегі енгізу/шығару бағдарламалыры бар, олардың ішінде пернетақтадан оқу, ақпаратты экранға шығару, деректерді дискіден/ге енгізу/шығару және т.б. Егер BIOS-та қате табылса, онда оны амалдық жүйе түзей алады.

Компьютер алғаш қосылғанда BIOS жіберіледі де келесі қадамдар орындалады:

1) BIOS орнатылған оперативті жадтың мөлшерін, пернетақта және басқа негізгі құрылғылардың қосылып тұрғандығын және олардың берген жауаптарының дұрыстығын тексереді. Тексеру ISA және PCI құрларынан басталады, себебі оларға барлық негізгі құрылғылар қосылады. Қосылған барлық құрылғылар, солардың ішінде plug and play құрылғысы да, тіркеледі;

2) Егер компьютердегі құрылғылар оның соңғы жүктелгенінен өзгеше болса, онда жаңа құрылғылар конфигурацияланады;

3) BIOS қай құрылғыдан (иілгіш дискіден, компакт дискіден, қатты дискіден) жүктелу керектігін анықтайды, ол үшін CMOS жадында болатын тізімді қарайды. Пайдаланушы бұл тізімді өзгерте алады;

4) Жүктелу құрылғысынан жадтқа жазылғанның бірінші секторы орындалады, себебі, онда жүктелу секторының аяғында орналасқан қандай бөлім белсенді болатындығы көрсетілген кестені оқитын бағдарлама болады;

5) Сол бөлімнен екінші жүктегіш оқылады. Ол амалдық жүйені белсенді бөлімнен оқып, оны жібереді;

6) Амалдық жүйе компьютердің конфигурациясын білу үшін BIOS-қа сұрақ қояды;

7) Ол тіркелген барлық құрылғылардың драйверлері барлығын тексереді;

8) Егер драйвер болмаса, онда амалдық жүйе пайдаланушыдан қажет драйвер бар иілгіш немесе компакт дискіні компьютерге қоюды сұрайды;

9) Егер драйвер орнында болса, онда амалдық жүйе оны ядроға жазады;

10) Амалдық жүйе драйверлер кестесін айқындайды;

11) Амалдық жүйе барлық аялық (фондық) процестерді жасайды;

12) Әрбір терминалда құпиясөзді енгізетін бағдарламаны немесе графикалық интерфейсті жібереді.

Негізі түсінік бойынша нақты Амалдық жүйенің құрларының түрлерін және олардың жұмыс ітеу принциптерін білмейінше, онымен жұмыс істеу мүмкін емес және қазіргі дербес компьютердің кез келген пайдаланушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

✓ Дәріске әдістемелік нұсқау

Дәрісте берілген материалды игеру үшін Амалдық жүйенің құрларының түрлері, олардың ерекшеліктеріне көңіл бөлу керек. Негізгі ұғымдарға анықтама беріп, оларды ажырата білу керек. Дәрісте берілген негізгі мәселелер мен сұрақтарға мән беріңіз. Берілген терминдерге анықтама беріңіз. Амалдық жүйенің құраларын талдау жасаңыз.

2.3 Амалдық жүйеде жадты басқару

Мақсаты:

Амалдық жүйеде жадты басқару проблемалары мен әдістерімен таныстыру.

Дәріс жоспары:

3. Адрестік кеңістік пен оперативті жад арасындағы байланыс;

4. Амалдық жүйеде жадты басқару әдістері.

Тақырып бойынша негізгі түсінік:

Адрестік кеңістік пен оперативті жад арасындағы байланыс.

Әрбір компьютерде орындалатын бағдарламаны сақтауға арналған оперативтік жад бар. Отырапайым амалдық жүйеде уақыттың нақты мезгілінде оперативтік жадта тек қана бір бағдарлама болады. Онда екінші бағдарламаны жіберу үшін алдымен бірінші бағдарламаны жадтан жойып алып, сонан кейін оның орнына екінші бағдарламаны жазу керек.

Дамыған амалдық жүйелер жадта бір мезгілде бірнеше бағдарламаның болғандығын қолдайды. Олар бір-біріне және амалдық жүйенің өзіне кедергі жасамас үшін қандай да бір қорғау механизмнің болғандығы қажет. Ол механизм аппараттық қамтамада орнатылады және амалдық жүйе арқылы басқарылады. Осыған байланысты амалдық жүйені жасар алдында оған компьютердің оперативті жадын басқаруға және оны қорғауға қатысты проблеманы шешіп алу қажет.

Келесі проблема маңызы жоғарыда айтылғаннан кем емес, бірақ процестердің адрестік кеңістігіне байланысты болады. Себебі, әрбір процеске пайдалануға болатын қандай да бір адрестер жиынтығы берілуі тиіс. Әдетте, ондай жиынтық 0-ден басталып қайсы бір максималды мөлшерге шейін аралықта жатады. Ең қарапайым жағдайда процестің адрестік кеңістігінің максималды шамасы негізгі жақтың көлемінен кіші болуы керек. Сонда процесс өзінің адрестік кеңістігін толтырады, ал жад оны бүтін қамтуына жеткілікті болады.

Бірақ біраз компьютерде адрес 32-разрядты немесе 64-разрядты болады. Бұл адрестік кеңістікке барлық байттар саны 2^{32} немесе 2^{64} болатындығын көрсетеді. Осы жерден өте күрделі сұрақ туады: *процестің адрестік кеңістігі компьютердің оперативті жад көлемінен көп болған жағдайда және осы процесс өзінің адрестік кеңістігін барлығын қолдануды талап етсе не істеу керек?* Алғашқы компьютерлерде бұл сұраққа жауап табылмады, сондықтан мұндай бағдарламаларды орындамауға тура келді. Бірақ бұл сұраққа жауап беру үшін компьютердің жадын басқару теориясы мен технологиялары пайда болды. Оларда берілген уақытта жақтың қандай бөлігі пайдаланып жатыр, қандай бөлігі босап жатыр, қандай процестерге қанша жад бөлігін беру керек, оперативті жад пен дискінің арасындағы деректер алмасын қалай орындау керек сияқты күрделі мәселелер шешілді.

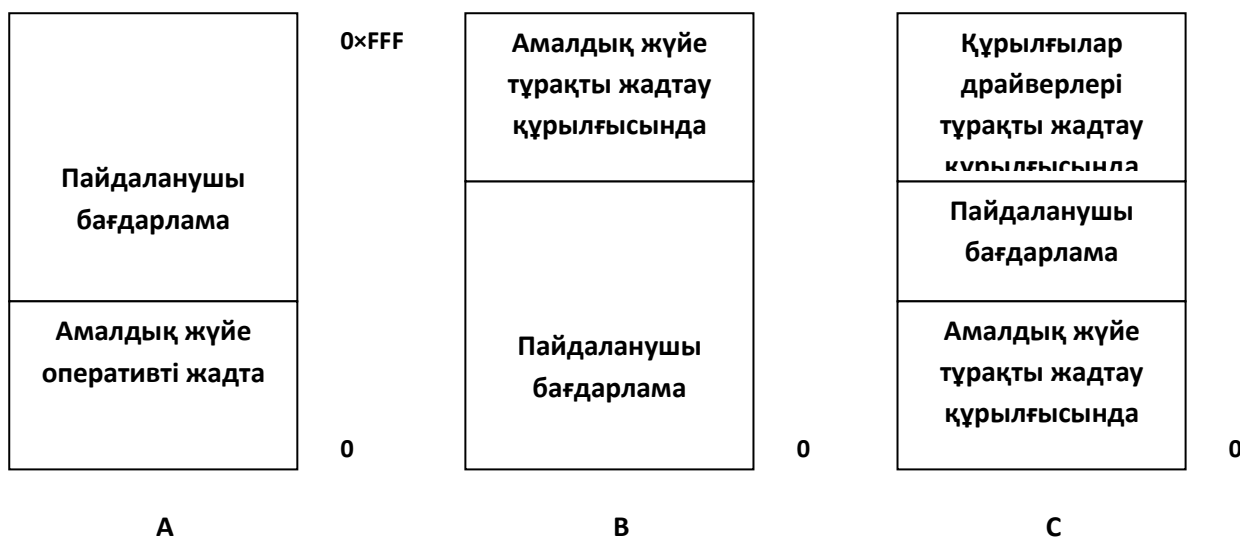
Амалдық жүйеде жадты басқару әдістері.

Жадты басқару әдістерін екі топқа бөлуге болады:

1) Процестердің орындалу кезінде оперативтік жад пен дискі арасындағы алмасып жазылуы. Мұнда процестерді толығымен басқылау (*swapping – полная подкачка*) немесе бетпен басқылау (*ragging – страничная подкачка*) жүзеге асырылады.

1) Дискіге басқылаусыз бір есепті жүйе.

Жадты басқарудың ең қарапайым әдісі ол уақыттың нақты мезгілінде бір ғана бағдарлама орындалуы және оперативті жадтың орындалатын бағдарлама мен амалдық жүйе арасында екіге бөлінуі. Төменгі суретте осының үш варианты көрсетілген.



A – амалдық жүйе жадтың төменгі жағындағы оперативті жадта, ал пайдаланушы бағдарламасы жоғарғы жақта болуы. Бұл модель бұрын мейнфреймде және мини-компьютерлерде қолданылды, бірақ қазір ол жалпы қолданыстан қалды.

B – амалдық жүйе жадтың ең жоғарғы жағындағы тұрақты жадтау құрылғысында, ал пайдаланушы бағдарламасы төменгі жақта болуы. Бұл модель кейбір қалталық компьютерлерде және ендірілген жүйелерде қолданылады.

C – құрылғылар драйверлері жадтың ең жоғарғы жағындағы тұрақты жадтау құрылғысында, пайдаланушы бағдарламасы жадтың ортасында, ал амалдық жүйе жадтың төменгі жағындағы тұрақты жадтау құрылғысында болуы. Бұл модель алғашқы (мысалы, MS DOS амалдық жүйесімен жұмыс істейтін) дербес компьютерлерде қолданылды. Мұндағы амалдық жүйенің тұрақты жадтау құрылғысындағы бөлігін BIOS деп атады.

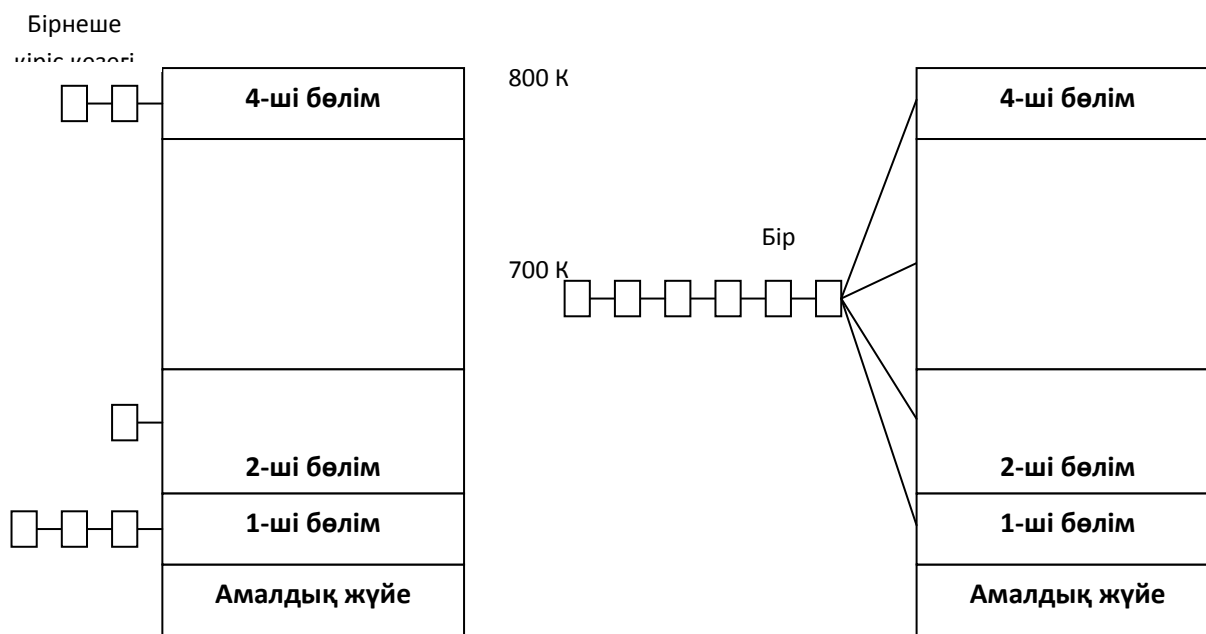
Компьютерде уақыттың нақты мезгілінде бір ғана бағдарлама орындалатын жағдайда пайдаланушы команданы тере салысымен амалдық жүйе сұратылатын бағдарламаны дискіден жадқа көшіріп, оны орындап, ал процесс аяқталған кезде экранға шақыру символын шығарып, келесі команданы күтеді. Команданы алып ол жаңа бағдарламаны жадқа бұрынғы бағдарлама үстіне жазады.

2) Бекітілген бөлімдерді пайдалатын көпесептілік

Көптеген жаңа амалдық жүйелерінің бір мезгілде бірнеше процестерді жіберу мүмкіндігі бар. Бір мезгілде бірнеше процестердің орындалуын бір процестің енгізі/шығару амалының аяқталуын күтуі, ал басқа процестің орталық процессорды пайдалануы деп түсіну керек. Көпесептілік процессордың жүктелуін өсіреді.

Көпесептілікке жетудің ең оңай жолы жадты (мүмкін тең емес) n бөлікке бөлу. Оны жүйені жіберген кезде қолмен жасауға да болады.

Тапсырма жадқа келгенде, оған кіру кезегіндегі осы тапсырма сиятындай ең кіші бөлікті тағайындау қажет. Себебі, бөліктердің көлемі өзгермейтін болғандықтан орындалатын процесс орналасқан бөліктің барлық бос орны пайдасыз құры қалады. Төменгі суретте көпесептіліктің бекітілген мөлшерлерлі бөліктерде орналасуы және жеке кіріс тапсырмаларның кезегі бар болуы көрсетілген.



A – үлкен A кке кезек жоқ, ал кіші бөлікке кезект B эп болған жағдайда жеке кезектегі кіріс тапсырмаларды сорттау тиімсіз екендігі түсінікті. Себебі кіші бөліктерге кезек көп, ал үлкен бөлік бос болуы мүмкін.

B – балама вариантта барлық бөліктер үшін бір ғана кезек ұйымдастырылған. Мұнда қайсы бөлік бұрын босатылса, оған жалпы кезектегі мөлшері жағынан сәйкес (бөлікке сиятын) ең жақын тапсырма орналасатын болады. Әйтседе, үлкен бөлікке кішкентай тапсырманы орналастырған тиімсіз болғандықтан, жалпы кезектен осы босатылған үлкен бөлікке мөлшері жағынан сәйкес ең үлкен тапсырманы таңдаған жөн.

✓ Дәріске әдістемелік нұсқау

Дәрісте берілген материалды игеру үшін амалдық жүйеде жадты басқару әдістерінің ерекшеліктеріне көңіл бөлу керек.

3 Бөлім. Файлдар жүйесі.

3.1 Файлдар

Мақсаты:

Файлдармен жұмыс істеу әдістерімен қарастыру.

Дәріс жоспары:

- 1 Файлдық жүйенің маңызы;
- 2 Файлдардың қызметі;
- 3 Файлдар түрлері.

Тақырып бойынша негізгі түсінік:

Файлдық жүйе барлық амалдық жүйелер қолдайтын өте маңызды ұғымның бірі. Бұрын айтылғандай, амалдық жүйенің негізгі функциясы дискілердің және басқа енгізу-шығару құрылғыларының ерекшелерін жасырып, пайдаланушыға түсінікті және ыңғайлы құрылғыға байланыссыз абстрактылы моделін көрсету болып табылады.

Файлдарды құру, жазу, оқу, немесе жою үшін жүйелік шақыруларды пайдалану қажет. Файлды оқыр алдында оны дискіге орналастырып және ашып, оқығаннан кейін оны жабу керек. Осы функциялардың барлығы операциялық жүйенің жүйелік шақырулары арқылы жүзеге асырылады.

Амалдық жүйелер файлды сақтау үшін орын тағайындағанда файлдарды топтау әдістері ретінде каталогтар (бумалар) түсінігін пайдаланады. Мысалы, студент оқу бағдарламасы үшін әрбір оқу курсына бір каталог, электрондық поштаға бір каталог, өзінің веб-беттеріне бір каталог және т.б каталогтар құруына болады. Каталогтарды жасау және жою үшін де жүйелік шақырулар керек. Олар бар файлдарды каталогқа салуды және одан шығаруды қамтамасыз етеді. Каталогтардың ішінде басқа каталогтар болуы мүмкін. Олар көпдеңгейлі қабаттасқан құрылым құрады.

Процестер мен файлдар иерархиясы бұтақ тәрізді болады. Онда бірнеше деңгей болады. Әдетте, процесс иерархиясының деңгейі онша терең емес, онда үш деңгейден көп болу өте сирек кездеседі. Ал файл құрылымы өзіне жетерліктей 4 немесе оданда көп деңгейлерді қолдана алады. Процесс деңгейі аз көп өмір сүрмейді, ереже бойынша, олардың өмір сүру уақыты бірнеше минут қана. Ал файлдар (каталогтар) иерархиясы бірнеше жылдар бойы сақталуы мүмкін.

Процестер мен файлдарға арналған тән болу және қорғау ережелері бөлек болады. Әдетте, тек аталық процесс ғана басқара алады немесе балалық процеске қолжеткізеді. Ал файлдар мен каталогтарды қарап оқу механизмі тек қана олардың иелеріне ғана беріліп қоймай, сонымен қатар жұмыс тобы мүшелеріне де пайдалануына мүмкіндік береді.

Әрбір файлды деңгейлі каталогта жол аттарын беру арқылы, анықтап білуімізге болады. Бұл файлдың толық аты деп аталады. Жол аттары каталог құрылысының басынан бастап жазылады. Бұны каталогтың түбірі деп атайды. Осындай абсолюттік жол аттары каталог

тізімдерінен тұрады, себебі каталогтардың түбірі немесе тамыры файлға баруы тиіс. Көп деңгейлі каталогтарда файлға баратын жолды деңгейлер арасын еңкейген сызықтармен бөледі. Бірінші сызық каталогтың түбірінен кейін қойылады. MS DOS және Windows символдардың орнына еңкейген сызықты құрамдас бөлшектердің бөліндісіне қайтадан қолдана алады. Онда нәтижесінде бұл жол былай өрнектеледі: C /Faculty/ Professor/ Courses.

Әр уақыттың кезеңінде, әр процессте соңғы жұмыс каталогы бар, яғни еңкейген сызықтан бастамастан бұрын файлға жол іздейді. C /Faculty/ Professor– жұмыс каталогы болып табылады, ал Courses жолына пайдаланылануға керек файлдарды тауып беретін абсолюттік жол.

Процестер жүйелік шақырулар арқылы өздерінің каталогтарын өзгерте алады. Файлды оқымастан немесе жазбастан бұрын, оны бірінші ашу керек және осы уақытта енгізуге қолжеткізу рұқсатының бар екенін тексеру қажет. Егер қолжеткізу рұқсаты болса, онда жүйе кішігірім бүтін санды шығарады. Ол файлдың дескрипторы деп аталады және кейінгі амалдарда қолданылады.

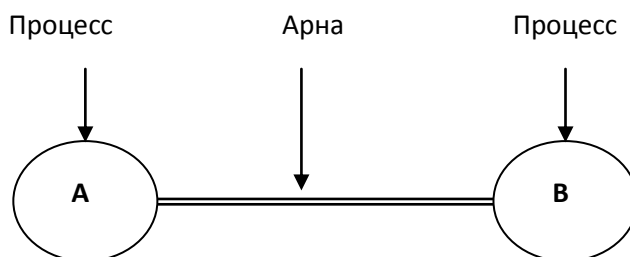
Егерде қолжеткізуге тыйым салынған болса, онда осыған байланысты қатенің коды экран бетіне шығарылады.

Амалдық жүйелеріндегі файлдық жүйе ендірілген болады. Барлық дерлік дербес компьютерлерде дискілерді салуға және алуға арналған бір немесе екі дискжетектер бар. Ауыспалы ақпарат тасымалдаушылармен (иілмелі диск, компакт-диск және т.б.) қарым-қатынас жасауға мүмкіндік беру үшін амалдық жүйе ауыспалы тасымалдаушының файлдық жүйесін негізгі бұтаққа қосады. Сонда жүйелік шақырудың алдындағы бір-біріне қатысы жоқ қатқыл дискідегі файлдық құрылым және ауыспалы тасымалдаушыдағы файлдық құрылым, жүйелік шақыру орындалғаннан кейін бір тамырдан (түбірден) үлкен иерархиялық құрылымның бұтақтары ретінде ғана болады. Осы механизм нақты енгізу-шығару құрылғылардың ерекшеліктерін ескермей ондағы деректермен абстрактылы файлдар құрылымы арқылы жұмыс істеуге мүмкіндік береді.

Амалдық жүйесіндегі маңызды ұғымның бірі арнайы файл болады. Арнайы файлдар енгізу-шығару құрылғыларын файлдар сияқты көрсетуіне көмек көрсетеді. Арнайы файлдардан ақпаратты оқу немесе жазу қарапайым файлдарға қолданылатын оқу немесе жазу амалдарын орындайтын жүйелік шақырулар арқылы жүзеге асады. Арнайы файлдардың екі түрі бар: *блокты файлдар*, *символдық файлдар*. Блокты арнайы файлдар еркін адрестелетін блоктар жиынтығынан тұратынын, мысалы, дискі, құрылғыларды моделдеу үшін қолданылады. Символдық арнайы файл символдар ағынымен жұмыс істейтін принтер, модем және басқа құрылғыларды моделдеу үшін қолданылады.

Амалдық жүйесінде файлдарға да процестерге де қатысты маңызды ұғымның тағы біреуі арна (pipe) болады. Арнаны (кейде құбыр деп

аталады) екі екі процесс бір-бірімен байланыс жасау үшін қажет псевдофайл деп түсінуге болады. Мысалы, төменде көрсетілгендей А және В процесі берілсе және олардың арасында қандай-да бір қатынас орнатылса, онда олардың бір-бірімен байланысы арнаның көмегімен жасалады.



Егер А процесі нақты деректерді В процесіне жіберетін болса, онда оларды арнаға жазады және мұнда арна шығу файлы ретінде қолданылады. В процесі арнадағы деректерді оқиды, ал мұнда арна кіріс файл ретінде болады. Сонымен, екі процесс арасындағы қарым-қатынас әдеттегі файлдарды оқу және жазу (енгізу-шығару амалдары) сияқты болады.

Негізі түсінік бойынша амалдық жүйедегі файлдармен жұмыс істеудің практикалық дағдыларын меңгермейінше, жалпы принциптерін білмейінше, жүйенің тиімді қолдануды үйрене алмайсыз және қазіргі дербес компьютердің кез келген пайдаланушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

➤ Дәріске әдістемелік нұсқау

Дәрісте берілген материалды игеру үшін файлдарға байланысты берілген анықтамаларына көңіл бөлу керек, негізгі ұғымдарға анықтама беріп, оларды ажырата білу керек. Дәрісте берілген негізгі мәселелер мен сұрақтарға мән беріңіз. Амалдық жүйенің файлдарына талдау жасаңыз.

2.2 Амалдық жүйе қауіпсіздігі

Мақсаты:

Қауіпсіздікті қамтамасыз ету және қолжеткізу құқығын орнату әдістерімен таныстыру.

Дәріс жоспары:

- 1 Қауіпсіздік түсінігі;
- 2 Қауіпсіздіктің есептері;
- 3 Қауіп және оның қолданылуы мен маңызы.

Тақырып бойынша негізгі түсінік:

Компьютерлер көп мөлшерлі маңызды ақпараттарды сақтайды. Олар ішінде басқалардан қорғауды талап ететін пайдаланушының электрондық хаттары, бизнес-жоспары, ғылыми еңбектері, жеке басына байланысты құпия мәліметтер және басқалар бар файлдардан тұрады.

Амалдық жүйелердің шешетін есептеріне осындай файлдардың қорғауын басқару кіреді. Мысалы, бұл файлдармен тек арнаулы қолжеткізу құқығы бар пайдаланушылар ғана жұмыс істей алуы керек. Ол үшін амалдық жүйеде қауіпсіздік жүйе болуы қажет.

Мысалы, UNIX жүйесінде файлды қорғау үшін оған 9-битті екілік код беріледі. Бұл қорғау коды әрқайсысы үш биттен тұратын үш өрістен тұрады. Бірінші өріс - пайдаланушыға; екінші өріс - жұмыс тобының басқа мүшелеріне; үшінші өріс – басқа қалғандарына арналған. Әрбір өрістегі 3 бит rwx (read-оқу, write-жазу, execute-орындау)-биттері деп аталады. Мысалы, қорғау коды мынадай rwx r-x --x түрде берілсе, онда пайдаланушы оқи, жаза және орындай алады, жұмыс тобы мүшелері тек оқи және орындай алады, ал басқалар тек қана орындай алады, бірақ оқи және жаза алмайды дегенді білдіреді, яғни, дефис таңбасы өзінің орны сәйкес амалға рұқсат жоқтығын көрсетеді.

Амалдық жүйе файлдарды қорғаудан басқа, көптеген басқа қауіпсіздік шараларын қамти алады. Онда адамнан, маңызды емес қонақтардан, зиянды қосымшалардан (тек қана вирустардан емес) қорғайтын компоненттер бар

Жалпы қауіпсіздік проблемасы өте күрделі және көп жақты болады. Оның үш маңызды аспектілері болып қауыптың табиғаты, зиянкестің табиғаты, деректерді кездейсоқ жоғалту табылады. Компьютерлік жүйенің төнген қауіптің түріне байланысты үш негізгі есебі бар, ол төменгі кестеде көрсетілген

№	Есеп	Қауіп
1.	Деректердің құпиялығы	Деректерді көрсету
2.	Деректердің бүтіндігі	Деректерді бүлдіру немесе өзгерту
3.	Жүйеге қолжеткізу	Қызмет көрсетуден бас тарту

Бірінші есеп - деректердің құпиялығы деген түсінік құпия деректердің құпия болуы керектігін білдіреді. Мысалы, кейбір деректердің иесі осы деректерге тек белгілі бір топқа кіретіндердің ғана қолжеткізуі болуы керек деп шешсе, онда амалдық жүйе осы топтың сыртындағыларға осы деректерге қолжеткізу мүмкін еместігіне кепіл беру керек. Мұнда деректер иесі амалдық жүйеге өзінің деректеріне қолжеткізу құқығы барлардың тізімін беру ғана қажет, ал амалдық жүйе осы талапты қатал орындауы керек.

Екінші есеп - деректердің бүтіндігі деген түсінік деректер рұқсатынсыз басқа айқындалмаған пайдаланушылар осы деректерді өзгертуге құқықтары жоқ және мүмкіндіктері мүлдем болмауы керектігін білдіреді. Деректерді өзгерту дегенге оларды жою немесе оларға жалған деректер қосу да жатады. Амалдық жүйе осындай рұқсатсыз амалдардан деректерді қорғауы қажет.

Үшінші есеп - жүйеге қолжеткізу деген түсінік компьютерлік жүйені ешкім істен шығара алмауы қажет екендігін білдіреді. Қызмет көрсетуден бас тарту деген шабуылдар өте кең тараған болып есептеледі. Мысалы, егер компьютер Интернет желіде сервер болса, онда ол өте көп сұратымдар ағынына ұшырап қалып қалуы мүмкін. Себебі оның уақытының барлығы келген сұратымдарды зерттеумен өтеді.

Осы айтылған есептерді шешу үшін көптеген моделдер мен технологиялар бар.

Амалдық жүйе жүйелік шақыруларды орындайтын бағдарлама. Редактор, компилятор, ассемблерлер, командалық интерпретаторлар амалдық жүйенің бөлімдері емес. Біз бұларды шатастырып алуымыз мүмкін. Сондықтан, амалдық жүйеде оларды шақыру үшін командалық қауіз (shell - оболочка) жасалынады. Ол амалдық жүйенің белгілі бір функцияларын пайдаланады. Мұндай қауіз пайдаланушы мен амалдық жүйе арасындағы интерфейс болады. Қандайда бір пайдаланушы жүйеге кірген кезде қауіз жіберіледі. Қауізге кіру үшін стандартты құрылғылар болып пернетақта мен монитор есептеледі. Осылардың көмегімен пайдаланушы командалық жолда тиісті командаларды теріп, орындай алады.

Негізі түсінік бойынша нақты амалдық жүйедегі қауіпсіздік проблемаларын шешу әдістерін меңгермейінше, деректерді қорғау мүмкін емес; жүйенің тиімді қолдануды үйрене алмайсыз және қазіргі дербес компьютердің кез-келген пайдаланушысында ерте ме, кеш пе пайда болатын проблемаларды шеше алмайсыз.

3.3 Ойлардың қайталанып келуі

Мақсаты:

Кибернетика ғылымын әр салада қарастырып, онымен жұмыс істеу, технологияның дамуына байланысты кибернетика ғылымының пайдаланушығы маңызын анықтау.

Дәріс жоспары:

1. Технологияның қалыптасуы;
2. Динамикалық процесс;
3. Кибернетика саласы және оның ғылымда атқаратын рөлі.

Тақырып бойынша негізгі түсінік:

Кибернетика - басқару жүйелерінің заңдылық құрылымдары мен басқару процестері туралы ғылым. Сонымен қатар, кибернетика компьютер туралы ғылым. Кибернетика бірнеше жылдар бойы пайда болған. Ол технологиямен және басқа да салалармен тығыз байланысты. Мысалы, бұрынғы кезде көліктің болмағанына байланысты, адамдардың ойларында жаяу жүру негізі қалыптасты, ал жаңа технологияны ойлап табу олардың жоспарларында болмаған. Дербес компьютерлер миллиондаған адамдар өздерінің жеке компьютерлері болуын армандағаннан шыққан жоқ. Қазір оларды шығару өте арзан

және технология қатты дамыған. Технологиялар жүйеге қандай әсер ететінін және біздің көзқарасымызды өзгертетінін жиі ұмытып кетеміз.

Технология дамығанда кейбір идеялар ескіреді. Бірақ технологиялардағы басқа өзгерістер ұмытылып кеткен идеяларды қайта жаңғыртады. Бұл өзгерістер әсіресе жүйенің әртүрлі бөліктерінің өнімділіктеріне қатысты болады. Мысалы, орталық процессордың жылдамдығы жадтың жылдамыдығынан едәуір артық болса, онда кэш-жад өте көкейкесті болады. Егер жаңа технологиялар жадтың жылдамдығын процессор жылдамдығынан асырып жіберсе, онда кэш-жадтың керегі болмай қалады. Ал тағы біраз уақыт өткеннен кейін технология тағы да процессордың жылдамдығын қатты өсіріп жіберсе, онда қайтадан кэш-жад қажет болып қалады. Осылай ескі ойлар, идеялар қайталанып жүре береді.

Кибернетика бірнеше жыл дамып келеді. Яғни, осындай нақты емес, кітаптардан, уақыт өте келе біз «ескірген» концепцияларды көреміз де жаңа заманға сай тиімді технология пайда болды. Технологиядағы өзгерістер кейбір ескірген түсініктерді қайтадан әкеліп, қоса салады. Осы мәселеге байланысты түсінімімізді қарастырайық, не үшін концепция ескірген, өзгерілген түсініктерді қайтадан жаңғыртып, өндеп шығартады. Сол үшін көзімізді жеткізу керек болсын деген оймен, оның растығын білуіміз қажет.

Бұрын компьютер аппаратына командалар жиынтығы ендіру талап етілген. Сонан кейін командаларды бағдарлама арқылы интерпретациялайтын микробағдарламалау әдісі шықты да, команданы аппараттық жүзеге асыру ескіріп кетті. Осыдан кейін RISC-компьютерлер шығып микробағдарламалау (команданы бағдарлама арқылы интерпретациялау) ескіріп қалды. Себебі команданы аппараттық тура орындау өте жылдам болды. Қазір интерпретацияның қайта жаңғырып келе жатқанын байқап отырмыз. Ол Интернетпен берілетін Java-апплеттерге байланысты болып отыр. Олар қабылданғаннан кейін интерпретацияланады. Мұнда орындалу уақытының ешқандай маңызы жоқ, себебі, желіде ақпаратты жіберу және қабылдау процесі одан бірнеше есе көп уақытты талап етеді. Бірақ бір кезде жағдай керісінше өзгеріп кетуі мүмкін.

Бұрынғы компьютерлік жүйелер дискіге файлдарды көрші секторларға қатарымен орналастырған. Бұл әдіс өте қарапайым болған, бірақ икемді емес. Себебі, файл үлкейген кезде, оған орын жетіспей қалады. Файлдарды дискіде үздіксіз орналастыру, компакт-диск пайда болғанша, ескіріп қалған. Компакт-дискілерде файлдардың өсу проблемасы жоқ. Сондықтан файлдарды дискіде үздіксіз орналастыру қайтадан жаңғырып шыға келді. Қазіргі барлық компакт-дискідегі файлдық жүйелер осы әдіспен жазылған.

Ең соңында қарастыратынымыз - динамикалық байланыс. Бағдарлама жүйедегі қатені түзету үшін, кітапханалық процедуралар ауыстырылуы тиіс. Бұл нысанаға байланысты динамикалық байланыс өмірге

келді. Multicast жүйесін уақытша ойдан шығарып тастауға болады. Бірақ оны қайтадан ашып, амалдық жүйелерге тәсіл ретінде қолданылады. Бірнеше бағдарламаны кітапханалық процедураға бөлуге, бірақ ол үшін арнайы көшірмелер жасалады. Өйткені графикалық кітапханалар көлемі өте жылдам өседі. Қазіргі заманда көптеген амалдық жүйелер динамикалық байланысты қолдайды. Тізімді одан әрі жалғастырып, адамзаттың үлкен мысалын жазу керек. Олар: бүгінгі ойдың қалыптасуы бойынша ескі, ал ертең ол жаңа технологияны ашылып кетуіне себебші болуы мүмкін. Технология тек аппараттық жүйелерге ғана әсер етіп қоймай, бағдарламалық қамтамаға да әсерін тигізеді. 60-70 жылдары әліпбилі-цифрлы дисплей, электронды-сәулелі құбырлар, 25x80 шығару үшін тағайындау, көбіне механикамен қоректенетін құрылғылар терминдері болды. Графикалық терминалдар 1960 жылға дейінде пайдаланылған. Бірақ олардың әрқайсысы он мың долларға дейін бағаланды. Кейін баға күрт құлағанда ғана әрбір пайдаланушы үшін дербес терминалдар жасалынды.

Негізі түсінік бойынша жаңа идеялар технологиялардың дамуына байланысты ескіріп немесе қайтадан жаңғырып отырады, қазіргі дербес компьютерде осылардың әртүрлі жетістіктері қолданылғандығын талдаңыз.

3.4 Жүйелік шақырулар

Мақсаты:

Жүйелік шақырулармен жұмыс әдістерімен таныстыру және жүйелік шақырудың мазмұнын ашып таныстыру.

Дәріс жоспары:

1. Жүйелік шақыруға түсініктеме беру;
2. Шақырудың қолданылуы;
3. Шақырудың тиімділігі мен маңызы;

Тақырып бойынша негізгі түсінік:

Жүйелік шақырулар жиыны кез келген амалдық жүйенің негізгі компоненті болады. Сонымен қатар, жүйелік шақырулар пайдаланушы мен амалдық жүйе арасындағы интерфейсті анықтайды. Пайдаланушы интерфейсіндегі жүйелік шақырулар бір амалдық жүйеден екіншісіне өткен кезде өзгеріп отырады. Себебі, жүйелік шақырулар амалдық жүйенің нақты не істей алатындығын білдіреді, ал әртүрлі амалдық жүйенің мүмкіншіліктері әртүрлі болады. Соныдықтан амалдық жүйені жақсы түсіну үшін нақты амалдық жүйенің жүйелік шақыруларын білу қажет. Мысалы, UNIX амалдық жүйесінде файлдарды оқу үшін үш параметрлі жүйелік шақырулар қолданылады: бірінші – файлды беру үшін; екінші – файлдан оқылған деректерді қайда жазуды көрсету үшін; үшінші – оқылатын байттар санын көрсету үшін. Көптеген басқа амалдық жүйелерде осы сияқты жүйелік шақырулар бар, олардың бір-бірінен айырмашылығы өте көп емес, негізінен олар жүзеге асыру

механизмдеріне қатысты болады. Жүйелік функцияларға қатынасу механизмі машинаға тәуелсіз болғандықтан, олар ассемблер тілінде жүзеге асырылады. Әдетте, дайын жүйелік процедуралардың кітапханасы болады. Оларға C тілінде немесе басқа тілде жазылған бағдарламалардан қатынас жасауға болады.

Кез-келген бір процессорлы компьютер нақты бір уақыт мезгілінде бір ғана команда орындай алады. Егер процессор пайдаланушы режимінде пайдаланушының бағдарламасын орындаса және жүйелік қызметке мұқтаж болса, мысалы, файлдан деректерді оқу үшін, онда ол үзілім жасауы немесе амалдық жүйені шақыру үшін жүйелік шақырудың командасын орындауы керек. Сонан кейін, амалдық жүйе шақыру параметрлері бойынша шақырған процестің не істеуі қажеттігін анықтайды. Нәтижесінде, амалдық жүйе жүйелік шақыруды өңдейді де, басқаруды жүйелік шақырудан кейін тұрған командаға береді. Жүйелік шақыруды орындаудың өзі процедура шақыруын орындағанға ұқсас. Тек біріншісі жүйенің ядросына кіріп кетеді, ал екіншісі оны істемейді.

Жүйелік шақыру механизмін түсіну үшін файлды оқу (read) жүйелік шақыруды қарастырайық. Оның негізгі үш параметр бар: біріншісі - файлдың беру үшін, екіншісі - буферді анықтау, үшіншісі – оқылатын байттың санын анықтау үшін. Барлық жүйелік шақырулар сияқты ол жүйелік шақырудың атауымен кітапханалық процедурасының шақыруы арқылы C тіліндегі бағдарламадан жіберіледі.

Егер жүйелік шақыру орындалатын болса, онда ол нақты оқылған байттар санын қайтарады және буферге файлдың ішіндегісін жазады. Ал егер жүйелік шақыру қандай-да бір жағдайға, мысалы параметрлерінің дұрыс берілмеуіне немесе дискілік қатеге, байланысты орындалмайтын болса, онда қателер санағышына бір қосылады, ал қатенің нөмірі қателер тізіміне түседі.

Кітапханалық процедура ассемблерде жазылуы мүмкін. Әдетте, жүйелік шақырудың нөмірін амалдық жүйе күтетін жерге жазады, мысалы регистрге. Оны оқып опрациялық жүйе пайдаланушы режимінен ядро режиміне көшеді де жүйелік шақыруды орындай бастайды.

Сонымен, амалдық жүйе мен қолданбалы бағдарламалар арасындағы интерфейс жүйелік шақырулардың жиынтығы болады, ол амалдық жүйеге беріледі. Іс жүзінде амалдық жүйе не істей алатындығын түсіну үшін біз интерфейссті қарастырамыз. Жалпы, жүйелік шақырулар процес-терді басқару үшін, файлдарды басқару үшін, катологтарды басқару үшін және басқа қызмет көрсету үшін қолданылады. Олармен нақты амалдық жүйелерді оқыған кезде танысқан жөн.

4 Бөлім. Көпдеңгейлі жүйелер

4.1 Көпдеңгейлі жүйелер

Мақсаты:

Көпдеңгейлі амалдық жүйелерді қолданудағы тиімділігін қалай пайдалануды таныстыру.

Дәріс жоспары:

Көп процессорлы амалдық жүйелердің түрлерімен таныстыру.

Тақырып бойынша негізгі түсінік:

Көпдеңгейлі жүйеде бірнеше иерархиялық деңгей бар. Осындай құрылымы бар бірінші амалдық жүйе THE (Technische Hogeschool Eindhoven) деп аталды, оны голландиялық профессор Э.Дейкстра өзінің студенттерімен Electrologica X8 компьютеріне арнап жасады. Амалдық жүйеде 6 деңгей болды:

0-деңгей: процессордың уақытын бөлу және көпесептілік;

1-деңгей: жадты және магнитті барабанды басқару;

2-деңгей: оператор-процесс байланысы;

3-деңгей: енгізу-шығаруды басқару;

4-деңгей: пайдаланушы бағдарламалары;

5-деңгей: оператор.

0-деңгейде үзілім пайда болғанда немесе таймер қосылғанда процестерді айырып-қосу арқылы процессордың уақыты бөлінеді. Бұл деңгейден жоғарыда жүйелік процестердің тізбегінен тұратын, олардың әрқайсысын бір процессорда бірнеше процестер жіберілгендігін ойламай бағдарламалауға болатын. Демек, 0-деңгей процессордың базалық көпесептілігін қамтыды.

1-деңгей процестерге көлемі 512Ксөз болатындай қылып оперативті жадтың және магнитті барабанның кеңістігін бөледі. Бұл деңгейден жоғарыдағы процестер берілген мезгілде оперативті жадта немесе магнитті барабанда орналасқаны туралы қам жемейді және оның рограммалық қамтамасы беттің оперативті жадқа қажетінше түсіріп отыруды қадағалайды.

2-деңгей оператор тетігі мен процестер арасындағы байланысты басқаралы. Бұл деңгейден жоғарыдағы процестер өзінің оператор тетігі болды.

3-деңгей енгізу-шығару құрылғыларын басқарады және олардан шыққан және оларға кіретін ақпарат ағынын буферлейді. Бұл деңгейден жоғарыдағы процестер нақты енгізі-шығару құрылғының ерекшелігін ескерудің орнына пайдаланушыға ыңғайлы сипаты бар абстрактылы енгізі-шығару құрылғыларымен қатынас жасайды.

4-деңгейде пайдаланушы бағдарламалары орындалады. Олар процестер, жад, оператор тетігі, енгізі-шығару құрылғыларын басқару туралы қам жемейді, себебі, олардың барлығы бұдан төменгі деңгейде шешіліп қойған.

5-деңгейде жүйелік оператордың процесі орналасты.

Көпдеңгейлі амалдық жүйе концепциясы MULTICS амалдық жүйеде дамыды. Мұнда деңгейлер концентрлі сақиналармен жасалды және ішкі сақиналар сыртқыға қарағанда артықшылығы мол болды. Егер сыртқы сақинаның процедурасы ішкі сақинаның процедурасын шақырғысы келсе, онда ол шақыру орындалар алдында параметрлері мұқият тексерілетін жүйелік шақырудың баламасын орындау керек. Әрбір пайдаланушы процесі үшін MULTICS адрестік кеңістіктің бөлігі болса да, аппараттық жабдықтар деректерді қорғауды жадтың сегменттері деңгейінде қамтыды. Онда оқу, жазу немесе орындау үшін дербес процедураларға (жад сегменттеріне) қолжеткізуге рұқсат берілді немесе тиым салынды.

Айта кетсек, THE амалдық жүйесінде көпдеңгейлілік тек қана құрылымдық шешім болды, нәтижесінде деңгейлердің барлығы бір объектілі (машина тіліндегі) файлға байланды, ал MULTICS жүйесінде деңгейлерге бөлу механизмі орындалу кезінде аппараттық жабдықтармен қолдау тапты. MULTICS жолының артықшылығы көпдеңгейлілікті пайдаланушылық жүйелер құрылымына дейін кеңейтуге болатын мүмкіндікте жатыр. Мысалы, профессор студенттерді тестілеу және бағалау үшін бағдарлама жазып, оны n -ші сақинада жібере алады. Ал студенттік бағдарлама $n+1$ сақинада жұмыс жасайды, сондықтан олар өз бағаларын өзгерте алмайды.

Негізі түсінік бойынша көпдеңгейлі амалдық жүйемен жұмыс істеудің жалпы принциптерін білмейінше, жүйенің тиімді қолдануды үйрене алмайсыз және қазіргі дербес компьютердің кез-келген қолданушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

✓ Дәріске әдістемелік нұсқау

Дәрісте берілген материалды игеру үшін көпдеңгейлі амалдық жүйеде берілген анықтамаларына көңіл бөлу, оларды ажырата білу керек. Дәрісте берілген негізгі мәселелер мен сұрақтарға мән беріңіз. Берілген терминдерге анықтама беріңіз. Көпдеңгейлі амалдық жүйенің атқаратын қызметтеріне талдау жасаңыз.

4.2 Виртуалды машина

Мақсаты:

Виртуалды машинаның жұмыс істеу принциптерімен таныстыру.

Дәріс жоспары:

1. Виртуалды машинаның мүмкіншілігі мен қолдануы;
2. Ескі бағдарламалық қамтаманы жаңа компьютерде жүргізу проблемасы.

Тақырып бойынша негізгі түсінік:

Орталық процессор процесстерді орындаған кезде бір процесстен басқа процеске көшіп отырады, сондықтан әрбір процестің виртуалды

орталық процессоры бар деп алған процестердің қасиеттерін және жағдайларын түсінуге жеңілдік жасайды.

Алғашқы OS/360 тек дестелік өңдеуді қолдайтын ғана амалдық жүйесі болған. Бірақ көптеген пайдаланушылар онымен уақытты бөлу режимінде жұмыс істегісі келді. Сондықтан оған уақытты бөлу режимін қолдайтыг қосымша бағдарлама жасалып қосылды. Осыны ескеріп IBM компаниясы TSS/360 атты уақытты бөле амалдық жүйесін жасап шығарды. Бірақ ол өте үлкен және баяу жұмыс істегендіктен оған көп пайдаланушылар өтпей қойды. Кейін IBM компаниясының Массачусетс штатында орналасқан Кембридждегі ғылыми орталығының тобы TSS/360-дан түбегейлі айырмашылығы бар уақытты бөлудің мүлде жаңа CP/CMS деп аталған амалдық жүйені жасап шығарды. Кейін оның атауын VM/370 деп өзгертті. Бұл әлі күнге шейін қалған майнфреймдерде жұмыс істеуде.

VM/370 амалдық жүйесін жасау кезінде уақытты бөлу жүйесі көпесептілікті және нақты құрылғыларды беретін машиналарға қарағанда ыңғайлы интерфейсі бар кеңейтілген машинаны қамтитындығы ескерілді. Сондықтан VM/370 амалдық жүйесінде осы екі функция бір-бірінен толық ажыратылады.

Жүйенің түйіні виртуалды машина монитормы деп аталады, ол жоғарғы деңгейге бір емес бірнеше виртуалды машина беріп жабдықтармен жұмыс істейді және көпесептілікті қаматамасыз етеді. Оның басқа амалдық жүйелерден айырмашылығы, осы виртуалды машиналар кеңейтілген машиналар емес. Олар файлдарды және басқа ыңғайлықтарды қолдамайды және жалаң аппаратураның, ядро режимінің, деректерді енгізу-шығарудың, үзілімдердердің және нақты компьютерде бар басқалардың дәл көшірмесі болады. Әрбір виртуалды машина нақты жабдықтармен бірдей болғандықтан, олардың әрқайсысында аппаратурада тура жіберілетін кез келген амалдық жүйелер жұмыс істей алады. Әртүрлі виртуалды машиналарда әртүрлі амалдық жүйелер жұмыс істей алады. Мысалы, олардың бірінде дестелік өңдеуді ғана қолдайтын көп пайдаланушылық OS/360 жұмыс істесе, ал басқасында уақытты интерактивті бөлетін бір пайдаланушылық CMS (Conversational Monitor System – диалогтық өңдеу жүйесі) жұмыс істейтін болады.

CMS амалдық жүйесінің бағдарламасы жүйелік шақыруды орындағанда амалдық жүйе өзінің жеке виртуалды компьютерінде үзілім жасайды, VM/370 жүйесінде виртуалды машинада емес нақты машинада болған жағдай сияқты болады. Сонан кейін CMS осы жүйелік шақыруға қажет болатын өзінің виртуалды дискісінен оқитын енгізу-шығару командасын немесе басқа команданы береді. Бұл команда VM/370 жүйесімен қабылданып, нақты жабдықтарды моделдеу аясында орындалады. Көпесептілік және кеңетілген машина функцияларын толық ажыратқанда әрбір бөлік едәуір қарапайым және ыңғайлы болады.

Виртуалды машина идеясы кейін өзгертілген күде басқа амалдық жүйелерде жүзеге асырылды. Мысалы, Pentium компьютерінде немесе басқа 32-разрядты Intell процессоры компьютерде MS-DOS амалдық жүйесінде жасалған ескі бағдарламалармен жұмыс жасау үшін виртуалды машиналар қолданылады. Ол үшін Pentium процессорында виртуалды 8086 процессор режимі жасалды. Бұл режимде машина 8086 сияқты жұмыс істейді, бірақ оның бағдарламалық қамтамасы 8088-бен бірдей және ол көлемі 1 Мбайт болатын 16-разрядты адрестеуі бар жадты да қолдайды. Мұндай режимдер MS-DOS бағдарламаларын жіберу үшін Windows және басқа амалдық жүйелерінде де қолданылады. Бағдарламалар 8086 виртуалды режимінде жіберіледі. Бағдарламалар әдеттегі командаларды орындағанда, олар құрылғылармен тікелей жұмыс істейді. Бірақ бағдарламалар жүйелік шақыруларды жасау үшін амалдық жүйенің үзіліміне қатынас жасаса немесе деректерді тікелей енгізу-шығару жасағысы келсе, онда виртуалды машина мониторуна көшетін үзілім болады.

Виртуалды машинаны орнатудың екі әдісі бар:

1) MS-DOS амалдық жүйесінің өзі 8086 виртуалды машинаның адрестік кеңістігіне орналасады. Мұнда виртуалды машинаның монитору нақты 8086 машинасында болатындай етіп үзілімді MS-DOS амалдық жүйесіне кері қайтарады. Сонан кейін MS-DOS өздігінен енгізу-шығаруды орындағысы келсе, виртуалды машина монитору оны тартып алып өзі орындайды.

2) Виртуалды машина монитору бірінші үзілімді алып қояды да, енгізу-шығаруды өзі орындайды. Себебі ол MS-DOS-тың барлық жүйелік шақыруларын және әр үзілім не істей алатынын біледі. Бұл біріншіге қарағанда жылдам жұмыс істейді, себебі, енгізу-шығару үшін MS-DOS-ты жүктемейді. Бірақ, ол біріншіге қарағанда мүмкіншілігі аз, себебі, ол тек MS-DOS-ты дұрыс моделдейді, бірақ басқа амалдық жүйелерге байланысты ешнәрсе жасай алмайды.

Виртуалды машиналар Java бағдарламасын орындау үшін де қолданылады. Sun Microsystem корпорациясы Java тілін жасағанда, оған сәйкес виртуалды машинаның архитектурасында ойлап шығарды. Ол JVM (Java Virtual Machine) деп аталды. Java тілінің компиляторы JVM-де орындалатын код шығарады. Бұл кодты Интернет арқылы JVM бар кез-келген компьютерге жіберіп орындауға болады.

➤ Қорытынды шығару

Негізі түсінік бойынша нақты амалдық жүйемен жұмыс істеудің практикалық дағдыларын меңгермейінше, онымен жұмыс істеу мүмкін емес; жалпы принциптерін білмейінше, жүйенің тиімді қолдануды үйрене алмайсыз және қазіргі дербес компьютердің кез-келген қолданушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

5 Бөлім. Процестер мен ағындар.

5.1 Процестер жағдайы

Мақсаты:

Процестердің жағдайларын түсіндіріу.

Дәріс жоспары:

1. Процестің құрылымы және жағдайлары;
2. Процестің бір жағдайдан басқа жағдайға өту диаграммасы.

Тақырып бойынша негізгі түсінік:

Процесс деп команда санағышының, регистрлердің және айнымалылардың мәндері қосылып орындалатын бағдарламаны айтады. Бірақ, процестер мен бағдарламалар арасында айырмашылық бар. Процесс белсенділікті білдіреді, оның бағдарламасы бар, кіріс-шығыс деректері бар және жағдайы бар.

Процестің жеке өзінің команда санағышы және ішкі жағдайлары бар тәуелсіз объект екендігіне қарамастан басқа процестермен өзара әрекеттесу қажеттігі болады. Мысалы, бір процестің шығыс деректері басқа процестің кіріс деректері болуы мүмкін.

Әрбір процесс өзінің жұмыс жылдамдығымен сипатталады. Ол бағдарламаның салыстырмалы қиындығына, әрбір процеске берілетін процессордың уақытына байланысты. Процестердің салыстырмалы жылдамдықтарына байланысты бір процесс жіберулеге (жұмыс істеуге) дайын тұруы, ал оған қажет кіріс деректері әлі жоқ болуы мүмкін. Бұл жағдайда осы процесс кіріс деректерін алғанша бұғатталады.

Процестер әртүрлі жағдайларға байланысты бұғатталады. Мысалы, процесс күткен деректер жоқ болса, онда бұл процесс осы деректер пайда болғанша бұғатталады. Сонымен қатар, жұмысқа дайын және жұмыс істей алатын процесс амалдық жүйе орталық процессордың уақытын басқа процеске беруіне байланысты бұғатталады. Бұл екі жағдай бір-бірінен мүлде өзгеше: бірінші жағдай процестің ішкі проблемасына байланысты, ал екінші жағдай процессордың жетіспеуіне байланысты техникалық проблема. Сонымен процессордың үш жағдайы болуы мүмкін:

1) Жұмыс істеу: процесс орталық процессорды пайдаланып тұрған мезгіл;

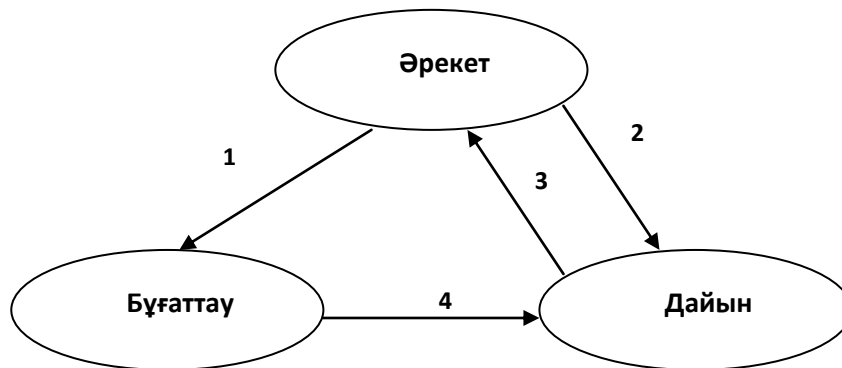
2) Жұмысқа дайын: процесс басқа процестің аяқталуын күтіп уақытша тоқталып тұр;

3) Бұғатталған: процесс қандай-да бір сыртқы оқиға болмайынша жіберіле алмайды.

Логика жағынан бірінші екі жағдай бірдей. Екі жағдайда да процесс жіберілуі мүмкін, тек екінші жағдайда процессор бос емес. Үшінші жағдай мүлде басқа, себебі онда процессордың жүктелгендігіне байланыссыз процесті жіберу мүмкін емес.

Жалпы орталық процессор бір процестен басқа процеске өту уақытын анықтайтын жоспарлау алгоритмін қолдану арқылы өтіп отырады. Оны амалдық жүйенің құрамына кіретін *процестерді жоспарлағыш* деген бағдарлама жүзеге асырады.

Төменде процестің жағдайлары үшін жағдайлар диаграммасы тұрғызылған, ондағы бағыттамалар жағдайлар арасындағы мүмкін болатын өтулерді көрсетеді.



1. Процесс кіріс деректерін күтіп бұғатталады.
2. Жоспарлағыш басқа процесті таңдады.
3. Жоспарлығыш осы процесті таңдады.
4. Кіріс деректеріне қолжеткізу бар.

Жоғарыда көрсетілген үш жағдайдың арасында төрт мүмкін болатын өтулер бар:

1-ші өту: процесс жұмыс істеу мүмкін емес екендігін байқағанда жүзеге асады. Кейбір жүйелерде процесс бұғатталған жағдайда болу үшін жүйелік шақыруды орындау керек. Мысалы, арнаны немесе файлды оқыған кезде кіріс деректері табылмаса процесс автоматты түрде бұғатталады.

2-ші және 3-ші өтулерді процестерді жоспарлағыш шақырады, сондықтан, процестердің өздері мұндай өтулер бар екендігін білмейді.

2-ші өту: процестерді жоспарлағыш орталық процессордың уақытын келесі процеске беру керектігін шешкенде жүзеге асады.

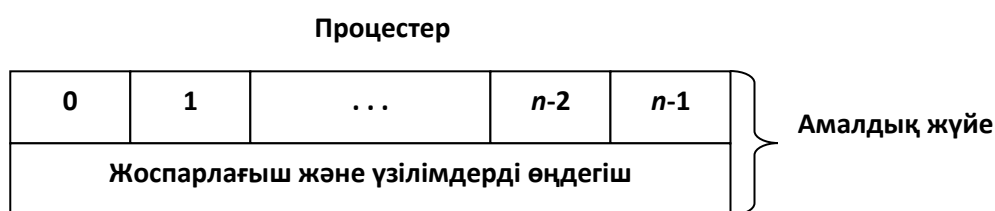
3-ші өту: барлық процестер өзінің процессорлық уақыттарын тамамдағанда, орталық процессор уақыты қайтадан бірінші процеске беріледі. Қай процесті қай уақытта және оған қанша уақыт беру керектігін жоспарлау өздігінен өте маңызды проблема. Осыны шешу үшін көптеген алгоритмдер жасалды. Оларда негізінен жалпы жүйе өнімділігі және жеке процестердің өнімділігі үшін талаптар арасындағы балансты анықтау мақсаты көзделген.

4-ші өту: процесс күткен сыртқы оқиға (мысалы, кіріс деректері келгенде) болғанда жүзеге асады. Егер бұл мезгілде қандай-да бір басқа процесс жіберілмесе, онда 3-ші өту жүзеге асып, процесс жіберіледі. Ал

оған кері жағдайда, орталық процессор босағанша процесс біраз уақыт жұмысқа дайын жағдайында бола тұруы қажет.

Процестердің моделі жүйенің ішкі іс-әрекетінің бейнелеуін оңайлатады. Кейбір процестер пернетақтадан терілген бағдарламаларды жібереді. Басқа процестер дискіні басқаруды жіберетін файлдық қызметтердің сұратымын орындау сияқты есептерді өңдейтін жүйенің бөліктері болады. Дискілік үзілім кезінде жүйе ағымдағы процесті тоқтатып, осы үзілімді күтіп тұрған дискілік процесті жібереді. Үзілім орнына біз күту уақытында бұғаттайтын дискілік процесті, пайдаланушы процесін, терминалдарды және т.с.с. қарауымызға болады. Оқиға болған, мысалы, ақпарат дискіден немесе пернетақтадан оқылған кезде бұғаттау алынады да, процесс жіберіледі.

Қарастырылған жағдай төмендегі суреттегі модельмен көрсетілген.



Амалдық жүйенің төменгі деңгейінде жоспарлағыш және үзілімдерді өңдегіш, ал жоғарғы деңгейінде процестер тізбегі орналасқан. Үзілімдерді өңдеу және процестерді жіберу мен тоқтатуға байланысты нақты әрекеттер жасырылған бағдарлама үлкен болмайды. Амалдық жүйенің басқа бөлігі процестер тізбегі түрінде ыңғайлы құрылым болып ұйымдастырылған.

Негізі түсінік бойынша процестің моделін және жағдайларын меңгермейінше, онымен жұмыс істеу мүмкін емес.

➤ Дәріске әдістемелік нұсқау

Дәрісте берілген материалды игеру үшін процестің моделіне және жағдайына берілген анықтамаларына көңіл бөлу керек, дәрісте берілген негізгі мәселелер мен сұрақтарға мән беріңіз. Процестің жағдайына талдау жасаңыз.

5.2 Ағындардың қолданылуы

Мақсаты:

Ағындардың қолдануын түсіндіру.

Дәріс жоспары:

1. Көп ағындарды қолдану қажеттігі

Тақырып бойынша негізгі түсінік:

Ағындардың бізге қажеттілігі көптеген қосымшалардың мезгіл мезгіл бұғатталатын біраз әрекеттерді орындауынан шығады. Қосымшаны квазипараллель режимде жіберілген бірнеше тізбектелген ағындарға бөліп бағдарламалық сұлбаны едәуір жеңілдетуге болады. Олар

адрестік кеңістікті және ондағы барлық деректерді бірігіп пайдаланады. Соңғы айтылған мүмкіншілік белгілі бір қосымшалар үшін өте маңызды, себебі, бұл жағдайда әртүрлі адрестік кеңістікті пайдаланатын параллельдік процестер жарамайды.

Ағындарды оңай жасап, оңай жоюға болады, себебі, олармен ешқандай ресурстар байланысты емес. Көпшілік жүйелерде ағынды жасауға процесті жасаудан 100 есе аз уақыт кетеді. Бұл қасиет ағындар санын жылдам өзгерту қажет болғанда өте пайдалы.

Ағындарды қолдайтын тағы бір себеп, ол өнімділік. Ағындар концепциясы өнімділіктің өсуін бермейді, егер олар процессордың мүмкіншіліктерімен шектелген болса. Бірақ, бір мезгілде көп көлемді есептеуді және енгізу-шығару амалдарын орындау қажеттілігі туса, онда ағындардың болуы осы әрекеттердің түрін уақыт бойынша қосып, олардың жалпы жұмысының өнімділігін арттыруға мүмкіндік береді.

Ең соңында, ағындар концепциясы көп процессорлы нағыз параллелизмі бар жүйеде өте пайдалы.

Ағындардың қажеттілігіне нақты мысал:

Мәтінді редактірлеуді аламыз. Пайдаланушы мәтінді редактірлеп терезеге шығарады. Олардың беттері ешбір өзгерусіз өз орындарында болады. Пайдаланушы өзіне керек емес жағдайда оны өзгертіп, өшіре алады.

Пайдаланушы кітап жазсын. Егер кітаптың авторы кітапты бір файлда сақтаған болса, онда керектіні іздеп табу және глобальды орын ауыстыру оңай болады. Дегенмен әрбір бөлімді жеке-жеке файлда да сақтауға болады. Бірақ файлдар саны көбейіп, орын ауыстыруға да, қоюға да, сонымен қатар түзеуге де өте қиын болады. Мысалы, кітапты басуға бергенде, автордың қалауы бойынша өзгерту керек болсын. Егер кітап бір файлда жазылса, онда оны бір командамен орындауға болады. Ал егер кітап 300 файлда жазылса, онда осы 300 файлды ашып 300 команда орындау керек болады.

Егер пайдаланушы көлемі 800 бет болатын құжаттың бірінші бетінен бір жолды алып тастаса не болатыны елестетейік. Пайдаланушы осы беттерді оқып, 600-ші беттегі сөйлемді өзгерткісі келсе, ол мәтіндік редакторға 600-ші бетке көшу туралы команда береді. Мәтіндік редактор барлық құжатты 600-ші бетке дейін форматтап шығады, себебі, форматтау алдында ол осы бет қай жерден басталатынын білмейді. Бұл едәуір уақытты алады.

Осы айтылғандарға ағындар көмектеседі. Мәтіндік редактор екі ағынды бағдарлама түрінде жазылған болсын. Бірінші ағын пайдаланушымен интерактивті әрекеттеседі, ал екінші ағын аялық режимде барлық құжатты форматтайды. Бірінші бетте сөйлем жойыла салысымен, интерактивті ағын аялық режимге құжатты форматтау туралы команда береді. Сол кезде бірінші ағын пернетақтадан команда алуын жалғастыра

береді. Екінші ағын құжатты форматтауын пайдаланушы өзінің жұмысын жасап отырып байқамай қалады.

Компьютердің кенеттен сөніп қалуына немесе басқа апаттық жағдайларға байланысты пайдаланушының біркүндік жұмысының нәтижесінен айырылып қалмауы үшін көптеген мәтіндік редакторлар бірнеше минут сайын мәтінді автоматты сақтап отырады. Осыған алдыңғы екі ағынды қатыстырмай, онымен үшінші ағын айналасады.

Бірақ осы жоғарыда айтылған жағдайларда үш процесті модель жарамайды. Себебі, үш процесс бір құжатпен жұмыс істеуі қажет. Ол мүмкін емес. Ал үш ағын бірге ортақ жадты пайдаланып бір құжатқа бір мезгілде қатынас жасауына болады. Мұнда жағдай басқа көптеген интерактивті бағдарламалар сияқты. Динамикалық электрондық кесте бағдарламасы пайдаланушыны матрицаның кейбір элементтерін толтыруға мүмкіндік береді. Оның қалған элементтері күрделі формулалар көмегімен орындалған есептеу арқылы толтырылады. Егер пайдаланушы бір элементте өзгеріс ендірсе, онда басқа элементтер автоматты түрде қайта есептеледі. Мұнда бірінші ағын аялық режимде қайта есептеумен айналасады, екінші ағын пайдаланушыға элементтерді өзгертуге мүмкіндік береді, ал үшінші ағын нәтижені мезгіл мезгіл сақтап отырады.

➤ Қорытынды шығару

Негізі түсінік бойынша ағындармен жұмыс істеудің практикалық дағдыларын меңгермейінше, онымен жұмыс істеу мүмкін емес; жалпы принциптерін білмейінше, жүйенің тиімді қолдануды үйрене алмайсыз және қазіргі дербес компьютердің кез келген пайдаланушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

➤ Дәріске әдістемелік нұсқау

Дәрісте берілген материалды игеру үшін ағындар туралы берілген анықтамаларына көңіл бөліп, оларды ажырата білу керек. Дәрісте берілген негізгі мәселелер мен сұрақтарға мән беріңіз. Ағындардың атқаратын қызметтеріне талдау жасаңыз.

5.3 Ядроға ағындарды іске асыру.

Мақсаты:

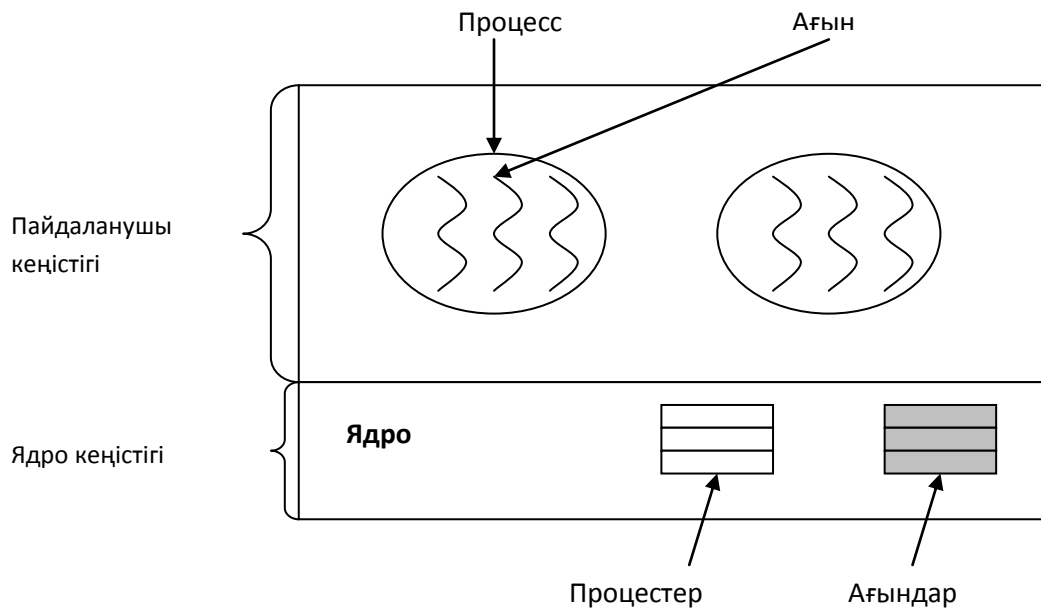
Ядроға ағындарды іске асыру әдістерімен таныстыру

Дәріске жоспар:

1. Ядро деңгейіндегі ағындармен жұмыс істеу;
2. Ағындарды ядро және пайдаланушы деңгейінде басқарудың аралас әдістері.

Тақырып бойынша негізгі түсінік:

Ядро ағындардың бар екендігін біліп, оларды басқарған жағдай қарастырайық. Мұндай жағдайда бағдарламалардың орындалуының қолдау жүйесін керек етпейді. Ол төменгі суретте көрсетілген.



Мұнда әрбір процесс үшін ағындар кестесінің қажеті жоқ, ол үшін жүйенің барлық ағындарын бақылап отыратын бір ғана ағындар кестесі бар. Егер ағын жаңа ағынды жасағысы немесе бар ағынды аяқтағысы келсе, онда ол ағындар кестесіне өзгерту ендіре отырып ағынды жасайтын немесе аяқтайтын ядроның сұратымын орындайды.

Ядродағы ағындар кестесі ағындардың регистрлерін, жағдайын және басқа деректерді қамтиды. Мұндағы ақпарат пайдаланушы деңгейінде ағындарды басқару сияқты болады, бірақ ол бағдарламалардың орындалуын қолдау жүйесінің ішіндегі пайдаланушының кеңістігінде емес, ядро кеңістігінде болады. Бұл ақпарат ядро өзінің әрбір бірағынды процестері туралы сақтайтын ақпараттың ішкі жиыны, яғни, процесс жағдайларының ішкі жиыны болады. Қосымша ядро жүйенің барлық процестерін бақылау үшін процестер кестесін қамтиды.

Ағынды бұғаттайтын барлық сұратымдар жүйелік сұратымдар сияқты жүзеге асырылады. Олар бағдарламалардың орындалуын қолдау жүйесінің процедураларды шақырудан едәуір көп уақытты талап етеді. Ағын бұғатталған кезде, ядро өз еркімен осы процестің басқа ағынын (егер орындалуға дайын басқа ағындар болса) немесе басқа процестің ағынын жібереді. Ағындарды басқару пайдаланушы деңгейінде болғанда, ядро процессорды басқа процеске бермейінше немесе орындалуға дайын жағдайдағы ағындар бітпейінше, бағдарламалардың орындалуын қолдау жүйесі бір процестің ағындарын жібереді.

Ядрода ағындарды жасау және аяқтау көп ресурсты талап ететін болғандықтан, кейбір жүйелер ағындарды қайта пайдаланады. Ағын аяқталғаннан кейін, ол жұмыс істемейді деп белгіленіп ажыратылады, бірақ оның ядрода сақталатын басқа деректер құрылымы өзгермейді.

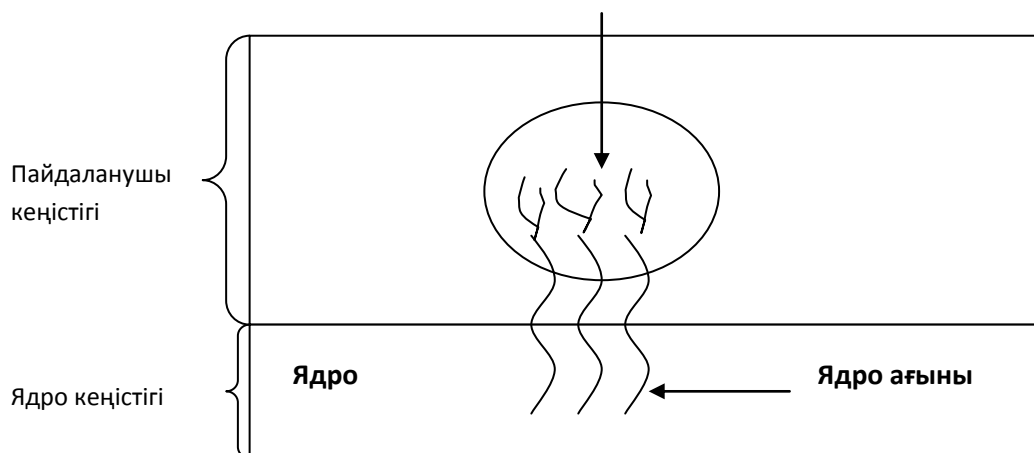
Кейін осындай жаңа ағынды жасау қажет болғанда, онда ажыратылған ағын тек қайтадан белсендіріледі де, оны жаңадан жасауға кететін шығын үнемделеді.

Пайдаланушы деңгейінде ағындарды басқарғанда да ағындарды қайта пайдалануға болады, бірақ мұнда ағындарды жаңадан жасауға кететін шығын аз болады, сондықтан ағындарды қайта пайдаланудың үлкен маңызы жоқ.

Ағындарды ядрода басқару қосымша жүйелік сұратымдарды талап етпейді. Сонымен бірге, егер бір ағын жад бетінің жоқтығы себепті қате шақырса, онда ядро басқа орындалуға дайын ағынның бар екендігін тексеріп, олар бар болғанда, талап етілген жад беті дискіден оқылғанша олардың біреуін жібереді. Ағындарды ядрода басқарудың кемшілігі, ол жүйелік сұратымдар бағасының көптігінде. Сондықтан ағындармен істелетін тұрақты жұмыс (жасау, аяқтау және т.б.) қосымша шығынды өсіреді.

Ағындарды ядро деңгейінде және пайдаланушы деңгейінде басқарудың артықшылықтарын ескеріп көптеген аралас әдістер пайда болды. Олардың біреуі ағындарды ядрода басқаруды қолданып, кейін пайдаланушы деңгейінде ағындарды мультиплексирулеу болып табылады. Ол төменгі суретті көрсетілген.

Ядродағы пайдаланушы ағындарын мультиплексирулеу



Ағындардың бұл моделінде ядро тек өз деңгейіндегі ағындарды ғана біледі және оларды басқарады. Осылардың кейбіреулері өздерінің үстінен мультиплексируленген пайдаланушылар деңгейіндегі бірнеше ағындарды қамтуы мүмкін. Пайдаланушылар деңгейіндегі ағындар көпағындылықты қолдамайтын жүйеде жіберілген пайдаланушылар деңгейіндегі процестің ағындары сияқты басқарылады. Өрбір ядро ағынында оны кезекпен қолданатын пайдаланушы деңгейінде ағындар жиынтығы бар.

Негізі түсінік бойынша ядродағы ағындарды іске асыру әдістерін меңгермейінше, онымен жұмыс істеу мүмкін емес.

➤ Дәріске әдістемелік нұсқау

Дәрісте берілген материалды игеру үшін ядродағы ағындарды іске асыру әдістерінде берілген анықтамаларына көңіл бөліп, оларды ажырата білу керек. Дәрісте берілген негізгі мәселелер мен сұрақтарға мән беріңіз. Берілген терминдерге анықтама беріңіз. Ядродағы ағындарды іске асыру әдістеріне талдау жасаңыз.

5.4 Ағындар моделі

Мақсаты:

Ағындар моделімен таныстыру

Дәріске жоспар:

1. Ағындардың қажеттілігі;
2. Ағындар моделінің түрлері.

Тақырып бойынша негізгі түсінік:

Ағындардың қажеттілігі

Жалпы процесс моделі екі тәуелсіз концепцияға негізделген: *ресурстарды топтау және бағдарламаны орындау*. Кейде оларды ажырату қажет, сонда ағындар түсінігі пайда болады.

Бір жағынан, процесті туыс ресурстарды бір топқа жинау әдісі ретінде қарауға болады. Процесте бағдарлама мәтіні мен деректерді қамтыған адрестік кеңістік және басқа ресурстар бар. Ресурстарға ашық файлдар, балалық процестер, өңделмеген апаттың хабары, сигналдарды өңдегіш, тіркеу ақпараты және басқалар жатады. Ресурстарды процестерге біріктіріп басқарған оңай болады.

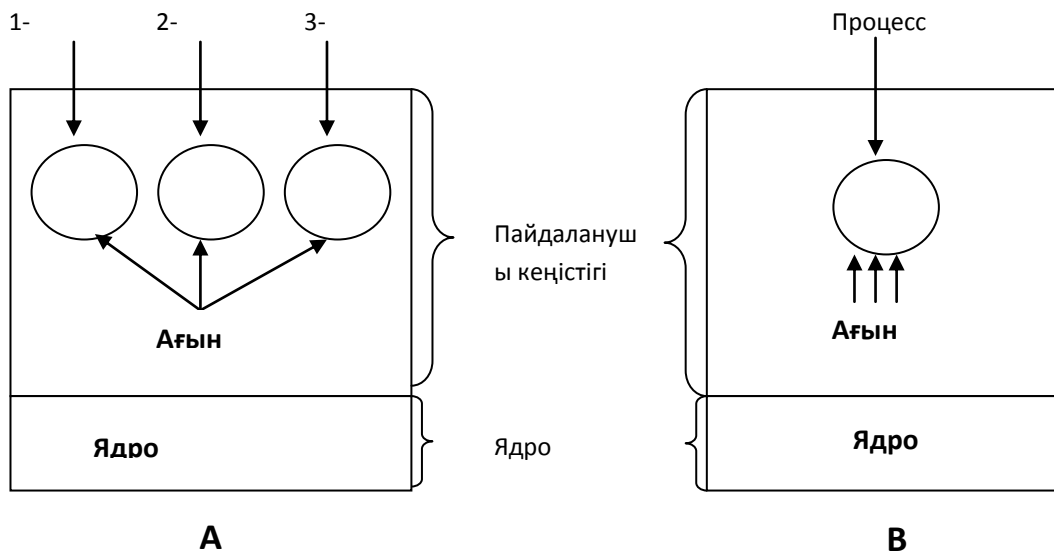
Екінші жағынан, процесті орындалатын командалар ағыны немесе жай ағын ретінде қарауға болады. Ағында әрекеттердің орындалу ретін бақылайтын команда санағышы бар. Оның ағымдағы айнымалылар сақталған регистрі бар. Оның процестің орындалу хаттамасын сақтаған стегі бар, онда әрбір шақырылған, бірақ қайтпаған процедураға жеке фрейм бөлінген. Ағын процестің ішінде атқарылғанмен, ағындар және процестер концепциясын ажырату керек. Процестер ресурстарды топтау үшін қолданылады, ал ағындар орталық процессорда кезекпен орындалатын объектілер ретінде қолданылады.

Ағындар концепциясы процестің моделіне тәуелсіздігі жеткілікті бірнеше бағдарламаларды бір ғана процестің ортасында бірізгіде орындау мүмкіншілігін қосады. Бір процессте параллель жұмыс істеген бірнеше ағын бір компьютерде параллель жүрген бірнеше процестерге ұқсас болады. Бірінші жағдайда ағындар адрестік кеңістікті, ашық файлдарды және басқа ресурстарды бөледі, ал екінші жағдайда процестер жадты, дискілерді, принтерлерді және басқа физикалық құрылғыларды бірге пайдаланады. Ағындар процестердің қарапайым

қасиеттеріне ие болады, сондықтан оларды кейде *оңайлатылған процестер* деп атайды. Көпәғындылық термині бір процессте көп ағынның пайдалануын сипаттайды.

Ағындар моделінің түрлері.

Төмендегі суретте процестер мен ағынның әрекеттесуі көрсетілген.



Мұнда А және В жағдайлары көрсетілген. Осы екі жағдайда да біз үш ағынмен жұмыс істейміз:

А – әрқайсысында өзінің адрестік кеңістігі бар және жеке басқару ағыны бар үш әдеттегі процесс көрсетілген.

В – үш ағыны бар бір процесс көрсетілген, бұл ағындар бір адрестік кеңістікті бөледі.

Әртүрлі ағындар бір процессте әртүрлі процестерге қарағанда онша тәуелсіз болмайды. Барлық ағындарда бір ғана адрестік кеңістік бар. Ол бірігіп глобальды айнымалыларды пайдалануды білдіреді. Кез келген ағын адрестік кеңістіктегі жақтың кез келген ұяшығына қолжеткізуі болғандықтан, бір ағын басқа ағынның стегінен ақпаратты оқи, жаза немесе өшіре алады. Қорғау болмайды, себебі (1) ол мүмкін емес (2) ол қажет емес. Мақсаттары бір-біріне сәйкес келмейтін әртүрлі пайдаланушылар жіберген әртүрлі процестерден ағындардың айырмашылығы, олар басында бір-біріне зиянын тигізбей бірге қосылып жұмыс істеу үшін жасалады. Төменгі кестеде процестер мен ағындар қолданатын элементтер келтірілген. (3-кесте)

3-кесте. Процестер мен ағындар элементтері

Процесс элементтері	Ағын элементтері
Адрестік кеңістік	Команда санағышы
Глобальды айнымала	Регистрлер
Ашық файлдар	Стек
Балалық процестер	Жағдайлар

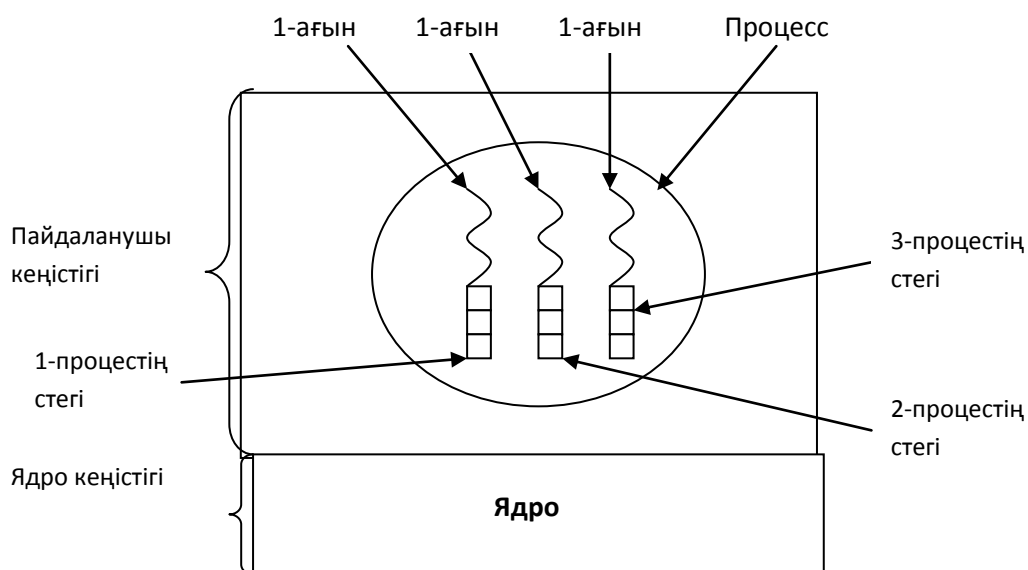
Бірінші бағанада процестің барлық ағындары бірігіп пайдаланатын элементтер келтірілген. Екінші бағанада әрбір ағынның жеке элементтері көрсетілген.

Бірінші бағанадағы элементтер тек процестің қасиеттері, олар ағынның қасиеттері емес. Мысалы, егер бір ағын файлды ашса, онда бұл файлды процесстегі қалған ағындар да көреді және олар ақпаратты одан оқи немесе оған жаза алатын болады. Бұл логикаға сәйкес келеді, себебі, ресурстарды басқару бірлігі болып есептелетін ағын емес, процесс. Егер әрбір ағында өзінің жеке адрестік кеңістігі, ашық файлдары, өңделмеген апат сигналдары және т.б. болса, онда ол жеке процесс болар еді. Ағындар концепциясы қандай-да бір есепті шешу үшін бірнеше ағындар өзара тығыз әрекеттесіп ресурстар жиынтығын бірге пайдаланудан тұрады.

Кез-келген әдеттегі (бір ағынды) процесс сияқты ағын бірнеше жағдайдың біреуінде бола алады: *жұмыс, бұғатталған, жұмысқа дайын, аяқталған*.

Жұмыс істеп тұрған ағын процессормен әрекеттеседі. Бұғатталған ағын өзін босататын қандай-да бір оқиғаны күтеді. Мысалы, пернетақтадан оқу жүйелік сұратымы орындалғанда пернетақтадан сигнал келгенше ағын бұғатталады. Ағын қандай-да бір сыртқы оқиға немесе басқа ағынмен босатылады. Жұмысқа дайын ағын өзіне кезек жеткенде жіберіледі. Ағын жағдайлары арасындағы өтулер процесс жағдайларындағы сияқты.

Төмендегі суретте көретілгендей, әрбір ағында өзінің стегі бар, ол әрбір шақырылған, бірақ басқаруды қайтармаған процедура үшін бір фрейм қамтиды. Фреймде процедураның локальды (ішкі) айнымалылары және қайту адресі болады.



Мысалы, егер X процедурасы Y процедурасын шақырса, ал ол, өз кезегінде, Z процедурасын шақырса, онда Z процедурасы жұмыс

істеген уақытта стекте барлық үш X, Y, Z процедураның фреймдері болады. Әрбір ағын әртүрлі процедураларды шақыра алады және осыған сәйкес, оның әртүрлі орындалу хаттамасы болады. Сондықтан, әрбір ағында өзінің жеке стегі болуы қажет.

Көпағынды режимде процестер, әдетте, бір ағынмен жіберіледі. Бұл ағын кітапханалық процедураларды шақырып, жаңа ағынды жасай алады. Мұнда жаңа ағынды жасау үшін жіберілетін процедураның аты шақырудың параметрі болады. Жаңа ағын бұрын бар ағынның адрестік кеңістігінде жасалатын болғандықтан, жаңа ағынның адрестік кеңістігі туралы ақпараттың керегі жоқ. Көпшілігінде барлық ағындар теңдәрежелі болып, иерархия құрылмайды. Бірақ, кейде «аталық ағын – балалық ағын» сияқты қатынастары бар ағындардың иерархиясы пайда болады. Иерархиялық қатынастарға байланыссыз жасайтын ағынға жаңа ағынның атын беретін идентификатор қайтарылады.

Тапсырманы орындап, кітапханалық процедураны шақырып, ағын жұмысын тоқтатуы мүмкін. Осыдан кейін ағын жойылады да, оны жоспарлағын іздемейтін болады. Кейбір ағындар жүйесінде бір ағын басқа (белгілі) ағынның аяқталуын күтеді. Ол үшін шақырушы процедураны басқа (белгілі) процедура аяқталғанша бұғаттайтын процедура шақырылады. Мұнда ағынды жаса мен аяқтау процесті жасау мен аяқтауға өте ұқсас.

Кейбір ағын өз еркімен өзінің кезегін басқа ағынға ұсынады. Бұл маңызды жағдай, себебі, ағындарда процестердегі сияқты уақытты бөлу режимін орнатуға мүмкіндік беретін таймер (уақыт) бойынша үзілім болмайды. Ағындарға сыпайы болу керек те, мезгіл мезгіл процессорды басқа ағынға беріп отыру қажет. Бір ағынды басқа ағын аяқталғанша күткізетін және оған басқа ағынның аяқталғандығы туралы ақпарат беретін процедуралар бар.

Ағындарға байланысты күрделі проблеманың бірі, ол ағындардың көп деректер құрылымын бірге пайдалануы. Оның маңызын түсіну үшін бірнеше нақты екі мысалды қарастырайық:

1) Егер бір ағын файлды, басқа ағын оны оқып жатқанда, жауып жібергенде не болады?

2) Егер бір ағынға жад жетіспей, ол қосымша жад сұрағанда, жарты жолда ағындар арасында ауыстырып-қосқыш істеп кетіп жаңа ағын іске қосылсын. Жұмыс барысында ол да қосымша жад сұраса, онда қосымша жад қатарынан екі рет бөлінген болар еді.

Негізі түсінік бойынша ағындар моделін меңгермейінше, онымен жұмыс істеу мүмкін емес; жүйенің тиімді қолдануды үйрене алмайсыз және қазіргі дербес компьютердің кез келген пайдаланушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

➤ Дәріске әдістемелік нұсқау

Дәрісте берілген материалды игеру үшін ағындар моделінде берілген анықтамаларына көңіл бөліп, оларды ажырата білу керек.

Берілген терминдерге анықтама беріңіз. Ағындар моделінің атқаратын қызметтеріне талдау жасаңыз.

5.5 Процестер және ағындар

Мақсаты:

Процестер мен ағындардың байланыстарымен таныстыру

Дәріс жоспары:

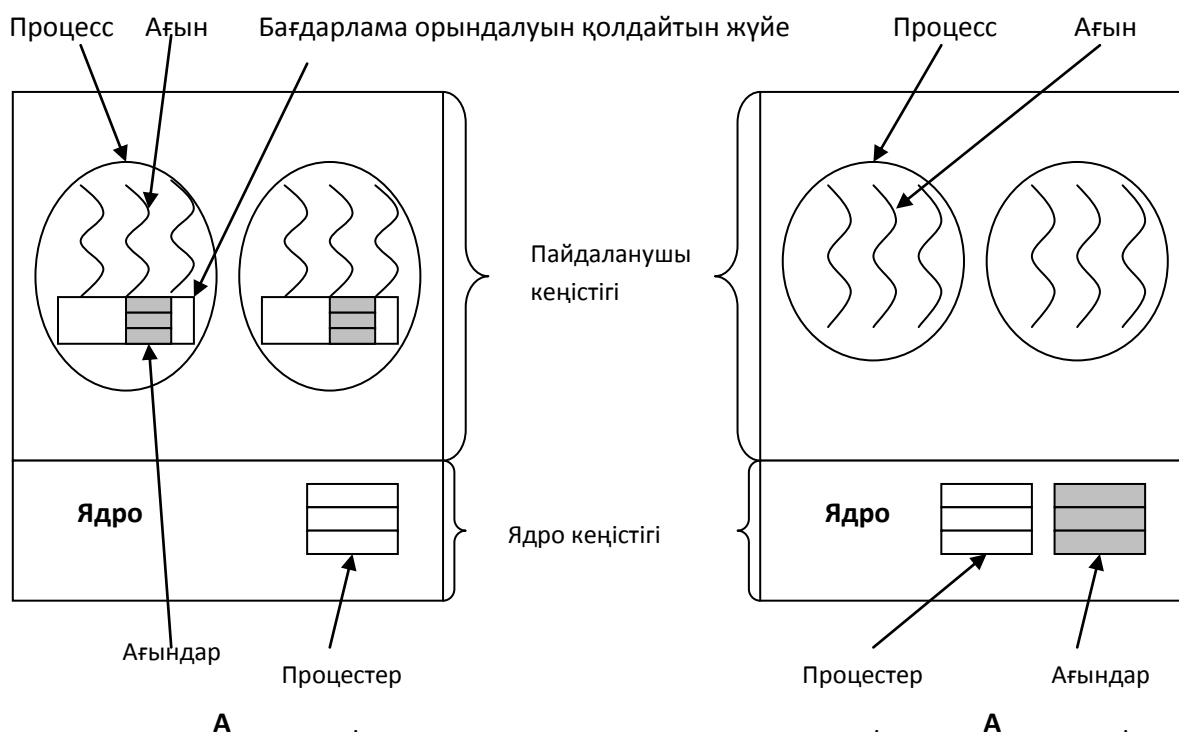
1. Процестер мен ағындардың байланыстары

2. Пайдаланушы кеңістігінде жүзеге асыру

Тақырып бойынша негізгі түсінік:

Амалдық жүйелерде әрбір процеске адрестік кеңістік және жеке басқарушы ағын сәйкес келеді. Расында осылар процесті анықтайды. Дегенмен, кейбір жағдайларда бір адрестік кеңістікте бірнеше квазипараллель басқарушы ағындар болғаны жөн. Олар бір адрестік кеңістікті бөліскен әртүрлі процестер сияқты көрінеді.

Процестер мен ағындардың байланыстарын түсіну үшін ағындардың пайдаланушы кеңістігінде қалай жүзеге асатындығын қарастырайық. Оның екі әдісі бар, олар төмендегі суретте көрсетілген.



А – пайдаланушы кеңістігіндегі ағын дестелері, В – ядромен басқарылатын ағын дестелері.

Бірінші әдісте ағындар дестесі түгелдей пайдаланушы кеңістігінде орналасады. Мұнда ядро әдеттегі бірағынды процесті басқарады да, ағындар дестесі туралы ешнәрсе білмейді. Ағындар ағындарды басқаратын процедуралар жиынтығы балатын бағдарлама орындалуын қолдау жүйесінің үстінен орындалады.

Бұл әдістің артықтышылығы, ол пайдаланушы деңгейінде ағын дестесін ағындарды қолдамайтын амалдық жүйеде де жүзеге асыруға болатындығында. Бұрын барлық амалдық жүйелер осы категорияға жататын, ал кейбіреулері кәзір де оған жатады.

Егер ағындарды басқару пайдаланушы кеңістігінде жүрсе, онда әрбір процессте өзіндегі ағындарды бақылайтын жеке *ағындар кестесі* болуы керек. Бұл кесте *процестер кестесіне* ұқсас болады, бірақ одан айырмашылығы, ол тек команда санағышы, стек төбесінің көрсеткіші, регистрлері, жағдайы сияқты ағындардың сипатын бақылайды. Ағын жұмысқа дайын немесе бұғаттау жағдайына өткенде қайта жүктеуге қажет барлық ақпарат, ядродағы процестер кестесінде сақталған процестер туралы ақпарат сияқты, ағындар кестесінде сақталады.

Ағын сол процесстегі басқа ағынның аяқталуын күткенде, ол локалды бұғаттауға әкелетін қандай-да бір әрекетті істеп, бағдарлама орындалуын қолдайтын жүйені шақырады. Процедура ағынды бұғаттау қажеттігін тексереді. Бұл жағдайда процедура ағынның регистрлерін ағын кестесінде сақтайды, ағын кестесінен жұмысқа дайын ағын іздейді және оның сақталған мәндерін машина регистріне жүктейді. Стек көрсеткіші мен команда санағышы ауыстырылып қосыла салысымен, жаңа ағынның жұмысы автоматты түрде жаңарады. Егер процессорда бір нұсқауда барлық регистрді сақтайтын команда болса және олардың барлығын бір жүктейтін тағы бір команда болса, онда ағындардың қайта қосылуы саны көп емес нұсқаулар арқылы орындалған болар еді.

Бірақ ағындардың процестерден өте күрделі айырмашылығы бар. Ағын өзінің жұмысын уақытша аяқтағанда, мысалы, ол X процедурасын шақырса, онда X бағдарламасы ағын туралы ақпаратты ағын кестесінде өзі сақтайды. Сонымен қатар, ол келесі ағынды таңдау үшін ағындар жоспарлағышын шақырады. Ағын туралы ақпарат сақтайтын процедура мен жоспарлағыш локалды процедура болғандықтан, олардың шақырулары ядроны шақырудан тиімді болады. Үзілімнің қажеті жоқ, контексті ауыстыру, кәшті сақтау және т.б. ағындарды ауыстырып қосуды едәуір жылдамдатады.

Пайдаланушы деңгейінде жүзеге асқан ағындардың басқа да артықшылықтары бар. Олар әрбір процеске өзінің жоспарлау алгоритмі болуына мүмкіндік береді. Кейбір қосымшаларға, мысалы, «қоқыс жинау» ағыны бар қосымшаларға жайсыз уақытта тоқтап қалатындығы туралы ойланбау ыңғайлы. Бұл қосымшалар жақсы масштабталады, себебі, ядро ағыны кестедегі және ядроның стегіндегі кеңістікті тұрақты толтыратындықтан, ағындардың саны көп болғанда үлкен проблемаға айналуы мүмкін. Ағынның пайдаланушы деңгейінде жүзеге асыруы өнімді болса да, оған байланысты кейбір проблемалар бар:

Бірінші проблема бұғатталған жүйелік шақыруды жүзеге асыруға байланысты. Мысалы, пернетақтада бір перне басылмас бұрын, ағын одан оқуды бастап кеткенде жүйелік шақыруды орындауға болмайды,

себебі ол барлық ағындарды тоқтатып қояды. Ағындарды қолданудың негізгі мақсаттарының бірі, ол әрбір ағынға бұғатталған ағын қалғандарына кедергі жасамайтындай етіп бұғаттау сұратымдарын пайдалануға рұқсат беру.

Екінші проблема бір ағынды жіберген уақытта басқа бір де бір ағын, бірінші ағын процессорды босатқанша, жіберілмейтіндігіне байланысты. Бір процестің ішінде таймер бойынша үзілім жоқ. Нәтижесінде кезекпен орындалатын ағындар жоспарлағышын жасауға болмайды. Жоспарлағыш, ағын өз еркімен бағдарлама орындалуын қолдау жүйесіне енбейінше, ешнәрсе істей алмайды.

Үшінші проблема бағдарламалаушылар ағындарды әдейі ағындары жиі бұғатталатын қосымшаларда қолданғысы келгендігі. Ондайларды, мысалы, көпағынды web-серверден табуға болады. Бұл ағындар жүйеге үнемі жүйелік сұратымдар жіберіп отырады. Басқаруды алып кеткен ядроға, жүйелік сұратымды жасау үшін, егер бір ағын бұғатталған болса, онда ағындарды ауыстырып қосуына ешқандай қиындық жоқ.

➤ Қорытынды шығару.

Негізі түсінік бойынша процестер мен ағындардың байланыстарын пайдаланушы кеңістігінде жүзеге асыруды меңгермейінше, онымен жұмыс істеу мүмкін емес және қазіргі дербес компьютердің кез келген пайдаланушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

➤ Дәріске әдістемелік нұсқау

Дәрісте берілген материалды игеру үшін процестер мен ағындардың байланыстарын пайдаланушы кеңістігінде жүзеге асыруда берілген анықтамаларына көңіл бөліп, оларды ажырата білу керек. Дәрісте берілген негізгі мәселелер мен сұрақтарға мән беріңіз. Процестер мен ағындардың байланыстарын пайдаланушы кеңістігінде жүзеге асыруға талдау жасаңыз.

5.6 Ағындар концепциясы

Мақсаты:

Ағындар концепциясын нақты қосымша жасау үшін қолдану әдістерін түсіндіру

Дәріс жоспары:

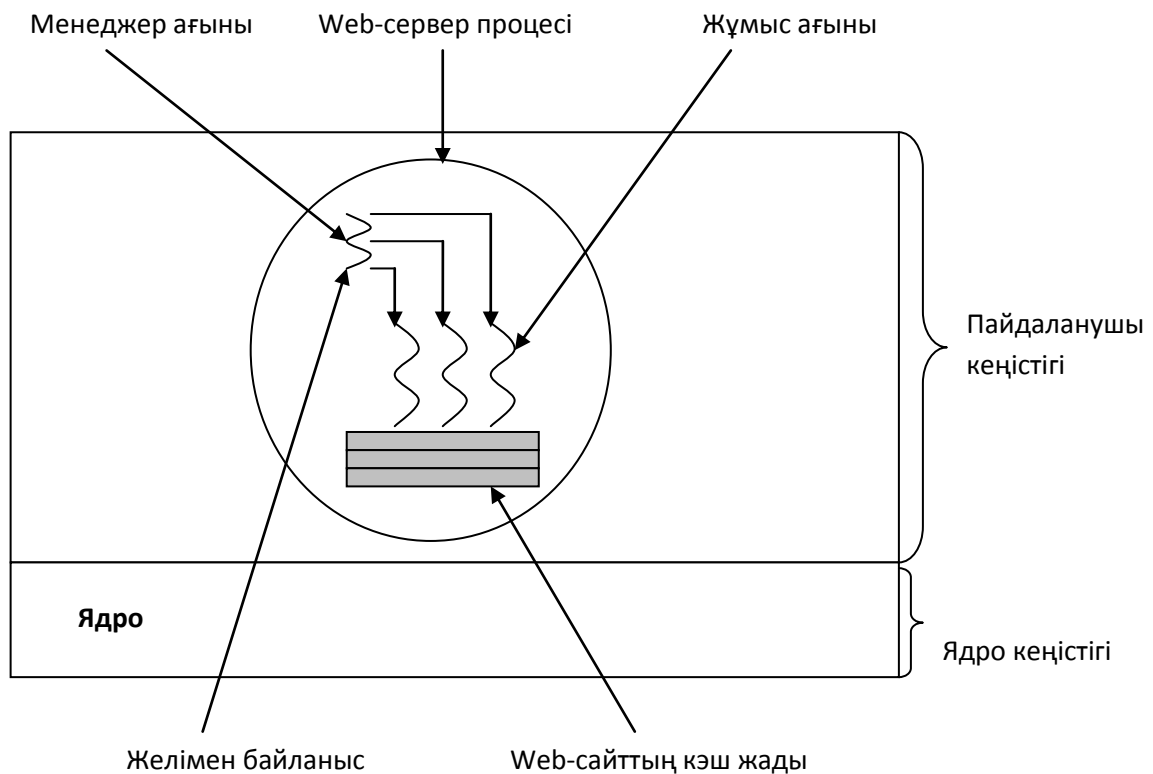
Ағындар концепциясын нақты қосымша жасау үшін қолдану

Тақырып бойынша негізгі түсінік:

Қосымша жасауда ағынның қажет екендігін дәлелдеу үшін бір мысал қарастырайық, ол Web-сайттың сервері.

Серверге көптеген сұратымдар келеді. Сервер сұратымға сәйкес Web-сайттың мазмұнын клиентке жібереді. Көпшілік Web-сайттарда кейбір беттерге басқасына қарағанда клиенттер көбірек қатысады. Мысалы, Sony компаниясының бас (негізгі) бетіне, нақты бейне

камералардың техникалық сипаттамасы бар бетке қарағанда, тұтынушылар едәуір жиі қарайды. Өзінің жұмысын тиімді ету үшін сервер осындай қасиеттерді пайдаланады. Ол үшін өте керек беттердің мазмұны негізгі жадтың бір бөлігінде сақталады да, дискіге жиі қатысудан құтылдырады. Жадтың бұл бөлігі *кэш* деп аталады. Ол басқа да жағдайларда қолданылады. Төменгі суретте Web-серверді ұйымдастырудың бір әдісі көрсетілген.



1-ші әдіс. Менеджер ағыны желіден келетін сұратымдарды оқиды. Сонан соң ол бос (яғни, бұғатталған) жұмысшы ағынын табады да, оған сұратымды береді және хабар нұсқағышын әрбір ағынға байланысты арнаулы сөзге жазады. Осыдан кейін менеджер күтіп тұрған ағынды бұғаттау жағдайынан жұмысқа дайын жағдайға ауыстырып белсендіреді.

Белсендірілгеннен кейін жұмысшы ағын Web-сайт сұратымының барлық ағынға еркін қатынауы бар кэш-жадында қанағаттандырылғандығын тексереді. Теріс жауап болған кезде ағын керек бетті дискіден оқу үшін оқу командасын орындай бастайды да, осы амал аяқталғанша бұғатталады. Жұмысшы ағын бұғатталғанда жіберу үшін басқа ағын таңдалады. Ол менеджер немесе дайын тұрған басқа жұмыс ағыны болуы мүмкін.

Бұл модель серверді тізбекті ағындардың жиынтығы ретінде жасауға болатындығын көрсетеді. Менеджер бағдарламасы құрамына сұратымды алау және оны жұмысшы ағынға беру амалдары бар ақырсыз циклдан тұрады. Әрбір жұмыс ағынының бағдарламасы

құрамында сұратымды менеджерден алу және сұратылған бет кэш-жадында бар екендігін тексеру амалдары бар ақырсыз циклдан тұрады. Егер сұратылған бет кэш-жадында бар болса, онда ол клиентке жіберіледі де, жұмыс процесі жаңа сұратымды күту үшін бұғатталады. Егер сұратылған бет кэш-жадында жоқ болса, онда ол дискіден оқылып, клиентке жіберіледі де, жұмыс процесі жаңа сұратымды күту үшін бұғатталады.

2-ші әдіс. Енді ағын жоқ кезде Web-серверді ұйымдастыруды талдайық. Бір мүмкін шешуі - Web-серверді бір ағын ретінде жұмыс істетуге. Негізгі цикл сұратымды қабылдайды, оны тексереді, қанағаттандырады сонан кейін келесі сұратымға кетеді. Мұнда керекті бет дискіден оқылып жатқанда, сервер тоқтап тұрады және басқа түсіп жатқан сұратымдарды өңдемейді.

Мұндағы Web-серверді ұйымдастырудың екі әдісінде де параллель бағдарламалау қолданылмайды. Бірақ, ағындарды қолданған бірінші әдіс, ағынсыз екінші әдіске қарағанда әжептәуір өнімділік береді.

3-ші әдіс. Жүйелік сұратымның бұғатталмайтын варианты бар болса. Серверге сұратым келгенде оны оқитын және тексеретін бір ғана ағын бар. Егер сұратылған бет кэш-жадында бар болса жақсы, ал егер жоқ болса, онда бұғаттаусыз дискілік амал жіберіледі.

Сервер кестеге сұратымның ағымдағы жағдайын жазып, келесі оқиғаға көшеді. Ол жаңа сұратым болуы немесе алдындағы амалдың жауабы болуы мүмкін. Жаңа сұратым жағдайында, сұратым өңделе бастайды, ал кері жағдайда сәйкес ақпарат кестеден оқылады да, жауап дайындалады. Бұғаттаусыз дискіден енгізі-шығару процедурасы жағдайында жауап сигнал немесе үзілім формасында болады. Бұл әдісте «тізбекті процестер» моделі жұмыс істемейді. Бағдарлама жағдайы, сервер сұратымдар арасында ауысып қосылып отырғанда, айқын түрде кестеге сақталуы және қайта қалпына келуі тиіс. Сонымен біз турасында ағындарды және стектерді моделдеп отырмыз. Мұндай әрбір есептеуге сақталған жағдай қарастырылған болып және осы жағдайларды өзгертетін бірнеше оқиғалар болса, онда оны ақырлы автоматты модель деп атайды.

Ағындар бұғаттаушы жүйелік сұратымдарды (мысалы, дискіден оқу) орындайтын тізбекті процестердің моделін сақтауға мүмкіндік бере тұра, параллелизмге қол жеткізеді. Бұғаттауы бар жүйелік сұратым бағдарламалауды жеңілдетеді, ал параллелизм өнімділікті өсіреді. Ақырлы автоматты модель параллелизм арқылы өнімділікті қатты көтереді, бірақ ол бағдарламалауды қиындататын бұғаттаусыз жүйелік сұратымды қолданады.

Егер бұғатталған жүйелік сұратымдар болса, процесс деректерді оқыған және жазған кезде бұғатталады. Мұнда процессордың тұрып қалуынан құтылу керек. Ол үшін ағындар концепциясын қолдану қажет.

Процесті кіріс ағыны, өңдеу ағыны жіне шығыс ағыны деп бөлі керек. Кіріс ағыны деректерді дискіден оқып, оларды кіріс буферіне жазады. Өңдеуші ағын деректерді кіріс буферден оқып, оларды өңдеп шығыс буферіне жазады. Ал шығыс ағыны деректерді шығыс буферінен оқып, оларды дискіге жазады. Бұл модельде деректерді оқу, өңдеу және жазу біруақытта өтеді. Әрине, ол жүйелік шақырулар бүкіл процесті емес, тек шақырушы ағынды бұғаттаса ғана жүзеге асырылады.

Негізі түсінік бойынша ағындарды нақты қосымшалар жасау үшін қолданудың практикалық дағдыларын меңгермейінше, онымен жұмыс істеу мүмкін емес және қазіргі дербес компьютердің кез келген пайдаланушысында ертеме, кешпе пайда болатын проблемаларды шеше алмайсыз.

➤ Дәріске әдістемелік нұсқау

Дәрісте берілген материалды игеру үшін ағындарды нақты қосымшалар жасау үшін қолданғанда берілген анықтамаларына көңіл бөліп, оларды ажырата білу керек. Дәрісте берілген негізгі мәселелер мен сұрақтарға мән беріңіз. Ағындарды нақты қосымшалар жасау үшін қолдануға талдау жасаңыз.

5.7 Процестің құрылуы

Мақсаты:

Процестерді құру әдістерімен таныстыру.

Дәріс жоспары:

Процестің құрылу тәсілдері.

Тақырып бойынша негізгі түсінік:

Компьютердегі барлық бағдарламалық қамтама, соның ішінде амалдық жүйеде, тізбектелген процестер жиыны түрінде ұйымдастырылған. Амалдық жүйеге барлық процестер бар екендігін растайтын әдіс керек. Қарапайым жүйеде және құрамында бір ғана қосымшасы бар жүйеде барлық ерте ме кешпе орындалатын процестердің болуын тексеру қиын емес. Әмбебап жүйелерде процестерді жасайтын және үзетін әдістерді ойлап табу керек. Жалпы процестерді жасау үшін келесі төрт оқиға болуы шарт:

- 1) Жүйені инициалдау;
- 2) Процесті жасау үшін жұмыс істеп тұрған процесс шығарған жүйелік шақыруды орындау;
- 3) Пайдаланушының процесті жасауға сұратымы;
- 4) Дестелік тапсырманы инициалдау.

Әдетте, амалдық жүйе жүктелген кезде бірнеше процессте жасалады. Олардың кейбіреулері жоғарағы қалаулы да, пайдаланушыны мен осы жұмысты орындаушының өзара әрекеттесуін қамтамасыз етеді. Қалғандары аялық болады, олардың нақты пайдаланушымен байланысы жоқ, бірақ олар ерекше функцияларды атқарады. Электрондық поштаға, Web-сайттарға, жаңалықтарға, баспаға шығаруға және басқаға

байланысты аялық процестер *демон* деп аталады. Үлкен жүйеде ондаған демондар болады.

Процестер тек амалдық жүйе жүктелгенде ғана жасалмайды, олар кейін де жасала береді, мысалы, ағындағы процестің сұратымы бойынша жасалады. Жаңа процестерді жасау, әсіресе, орындалатын есепті бір-бірімен байланысы бар, бірақ тәуелді емес, өзара әрекеттесетін процестер жиынтығы ретінде құрастырған жағдайда өте пайдалы. Егер ары қарай өңдеу үшін желіден үлкен мөлшерде деректер таңдау қажет болса, онда бірінші процесті деректерді таңдау және оларды бірге пайдаланатын буферге орналастыру үшін, ал екінші процесті буфердегі деректерді оқу және оларды өңдеу үшін жасау өте ыңғайлы болады. Мұнда, егер әрбір процесті көп процессорлы жүйедегі жеке процессорда жіберсе, онда бұл модель деректерді өңдеуді едәуір жылдамдатады. Интерактивті жүйеде бағдарламаны пайдаланушы пернетақтадан команданы теріп немесе мауспен оның бегісінде екі рет шертіп жібере алады. Екі жағдайда да нәтижесінде жаңа процесс жасалып, онда бағдарлама жіберіледі.

UNIX жүйесінде X Windows жұмыс істесе, жаңа процесс өзі жіберілген терезені алады. Windows жүйесінде процесс жіберілгенде, оның жеке терезесі болмайды, бірақ ол бір немесе бірнеше терезені жабуға міндетті. Айтылған амалдық жүйелердің екеуінде де пайдаланушы әрқайсына жеке процесс сәйкес келетін бірнеше терезелерді бірмезгілде аша алады. Пайдаланушы тышқан арқылы бір терезеден екінші терезеге өтіп, процестермен әрекеттесе алады. Мысалы, процестердің орындалуына қажет деректерді керегінше енгізіп отыруына болады.

Жаңа процесті жасауға әкелетін соңғы оқиға майнфреймдегі (үлкен компьютердегі) дестелерді өңдеу жүйесіне байланысты. Пайдаланушылар мүмкін қашықтағы терминал арқылы, дестелік тапсырманы жібереді, ал амалдық жүйе жаңа процесс жасап, қажетті ресурс босаған кезде кезектегі келесі тапсырманы жібереді.

Техникалық жағынан барлық жоғарыда айтылған жағдайларда жаңа процесс бірдей әдіспен жасалады: *ағымдағы процесс жаңа процесті жасау үшін жүйелік шақыруды орындайды*. Ағымдағы процестің рөлін пайдаланушы жіберген процесс, пернетақта немесе мауыс арқылы жіберілген жүйелік процесс, сондай-ақ, дестелерді басқаратын процесс орындауы мүмкін. Бірақ, барлық мүмкін жағдайда ағымдағы процесс тек жүйелік сұратымды орындап, жаңа процесті жасайды. Жүйелік сұратым амалдық жүйеге жаңа процесті жасаттырады және осы процессте жіберілетін бағдарлама жайында тура немесе көлденең ақпаратты қамтиды.

Процестердің жасалуы аяқталғаннан кейін, ол өзінің жұмысын бастайды. Бірақ ешнәрсе шексіз болмайтындықтан, процесс ерете ме кеш пе жұмысын аяқтайды.

6 Бөлім. Амалдық жүйелер

6.1 Windows NT/2000 амалдық жүйесі

.Дәріс мазмұны

1 Windows NT туралы жалпы мағлұматтар. Windows NT интерфейсі. Windows NT архитектуралық модулі.

2. Windows NT - нің жадысымен басқару. Основные отличия Windows 2000. NTFS4 (Windows NT) и NTFS5 (Windows 2000) файлдық жүйесі.

Windows NT 32 разрядты, алғашында 1993 жылы одан саң 1998 жылы жарыққа шыққан бірінші версиясы болып табылады. Бұл жүйе жоғары сенімді және тұрақты болып тек жұмыс жасауға арналған болатын. Сол себептен Windows NT-де түрлі қателіктер өте аз кездеседі. Бұның себебі NT OS/2 сияқты өзінің барлық программаларын басқаруын қадағалап отырады. Windows 3.1. 95/98/ME жүйелерінде әрбір іске қосылған программа өзін басты санап және жедел жадтағы кез келген орынға ие болу үшін күресті, сол себептен амалдық жүйенің қызметі нашарлады. NT бұл жағдайды кеңінен қарастырып әрбір программаға өзіне лайықты жедел жадтан орын береді. Бұл орындарда арнайы адрестер арқылы орнылысты. Бұдан бастап NT Windows 98 қарағанда компьютер ресурстарына кез келген программалардың мүмкіндігін шектеді. Бірақ NT - мен жұмыс жасаудан көптеген программалар бас тартты.

NT артықшылы жүйелік режимде жұмыс жасауда ғана, басқаша айтқанда кез келген компьютерлермен байланыс жасауда белгілі болады.

1998 жылдың аяғында Microsoft корпорациясы 1999 жылы шығуға болжаған

Windows NT 5.0. версиясы жаңа атпен, яғни, Windows 2000 шығады деп жариялады. Бірақ оның тек қана аты өзгеріп қоймай бұл амалдық жүйе тек бірлескен саудада емес, сонымен қатар үйдің дербескомпьютері болуы тиіс еді.

Windows 2000 амалдық жүйесі Windows NT 5.0 түрінде көрсетеді, бұл Window NT 4.0 жүйенің қасиетін алған. Бұл толығымен 32-разрядты көпесепті жүйе болып табылады.

Windows 98-ді басқару. Windows 98 командаларының көп бөлігі тышқанның көмегімен орындалады. *Тышқан нұсқағышы* деп аталатын басқару элементі тышқанмен тығыз байланыста болады. Тышқан нұсқағышының Жұмыс үстеліндегі козғалысын тышқанды жазық бетте козғау арқылы басқаруға болады. Жұмыс үстеліндегі кез-келген объектіге тышқанның оң жақ батырмасын шертсек *жанама меню* пайда болады. Жанама менюде осы объектімен жұмыс істеу кестесі көрсетіледі. Әр объект үшін жанама меню кестесі әр түрлі болады:

- қаптама үшін;
- таңбаша үшін.

Тышқанның көмегімен ашқарылапғын басқару тәсілдері:

шерту (тышқанның сол жақ батырмасын жылдам басып, жіберіпқалу);
оң батырманы шерту;
екі рет шерту;
сүйреу — сол жақ батырмасын басып тұрып, тышқанның орнынауыстыру;
тышқанды созу — сүйреу сияқты орындалады, бірақ бұл кездеобъектінің орны ауыспайды, оның өлшемдері өзгереді.
Жұмыс үстелінің негізгі элементтері.

Пиктограмма деп объектінің графитік бейнеленуін айтады. Пиктограммаға жасалған амалдар тікелей объектіге де қолданылады. Мысалы, пиктограмманы өшіру объектіні өшіру, пиктограмманы көшіргеніміз объектіні көшіру болып табылады.

Таңбаша (жарлық) — ол объектіге нұсқағыш қызметін атқарады. Таңбашаны өшірсек объектіні емес тек нұсқағышты өшіреміз. Таңбашамен жұмыс істеу тәсілдерінің пиктограммамен жұмыс тәсілдерінен еш айырмашылығы жоқ. Таңбашаның көмегімен кейбір жиі қолданылатын объектілерге жылдам қатынауға болады. Таңбашаның төменгі сол жақ бұрышында иілген жебелігі болады.

Есептер панелі — экранный ең төменгі жағында орналасқан сұр жолақ. Мұнда Іске қосу (Пуск) батырмасы орналасқан. Берілген уақыт мезетінде іске қосылып тұрған программалар есептер панелінің ортасында көрсетіледі. Ал оң жағында кейбір индикаторлар бейнеленген.

Іске қосу батырмасы. Мұнда Бас меню (Главное меню) орналасқан. Бұл батырма компьютердің кез-келген жеріне апаратын бас нүкте болып табылады.

Индикаторлар — кейбір мәліметгерді (уақыт, пернетақта режимі, принтердің күйі және т.б.) экранға шығарып қою үшін қажет.

Windows-тын файлдары мен қаптамалары.Файлдарды компьютердің қатқыл дискісіне сақтау тәсілдері *файлдыц жүйе* деп аталады. Барлық амалдық жүйелер сияқты Windows 98-де файлдық кұрылымды басқару кұралдарыш ұсынады.

Менің компьютерім (Мой компьютер) терезесін ашып, онда С: катқыл дискісінің таңбашасын тауып алыңыз. Сол таңбашаға екі рет шертсе жаңа терезе ашылады. Бұл терезеде С катқыл дискісінде орналасқан файлдар мен қаптамаларды көруге болады. Сол қаптаманың біреуін тандап алып, тышқан нұсқағышымен екі рет шерту арқылы оны ашып алуға болады.

Осы процесті қайталай отырып бағыныңқы қаптамалардың ең түпкі файлдарына жетуге болады.

Қаптама терезесі дегеніміз осы қаптаманың ішінде орналасқан файлдар мен қаптамалардың графитік түрде бейнеленуі. Кез-келген қаптама өз терезесінде ашылады. Осылайша бір мезгілде бірнеше терезені ашып қоюға болады.

6.2 UNIX амалдық жүйесі

Дәріс мазмұны

1. UNIX ОЖ негізгі компоненттері. UNIX операциялық жүйесінде пайдаланушының жұмысымен байланысты негізгі түсініктер. 2. Каталог және файлдар. Файлдың иесі және файлды қорғау. Мәтіндік файлдармен жұмыс. Стандартты файлдар. Программаны өңдеудің тәсілдері. Жүйелік администрациялау. Файлдық жүелер. *UNIX ОЖ тарихы.* 40-50-ші жылдары барлық компьютерлер дербес компьютерлер болатын, яғни ол кезде пайдаланушы компьютермен жұмыс жасауда белгілі сағаттарға ғана жазылатын және маши осы уақытта пайдаланушының мүмкіндігінде болатын. Осы компьютерлердің физикалық өлшемдері үлкен болатын, бірақ, осындай компьютерде жұмыс жасау әр уақытта бір ғана пайдаланушы болатын. 60-ші жылы осы машиналардың орнына пакетті жүйелер келді, мұнда, программист машиналық бөлмеге перфокарта түрінде тапсырмаларды әкеле бастады. Егер жеткілікті түрде тапсырмалар жиналса, онда оператор оларды бір пакетке магнитті лентаға көшіретін. Перфокарталарды өткізуден бастап баспаға шығыруға дейін бір немесе бірнеше сағат өтетін.

Осы бар схеманы барлығы өнімсіз және көңілдерінен шықпағаннан кейін, оны ары қарай іске асыруда, Дартмуттік колледжде және Массачусетском технологиялық институтта уақытты бөлу жүйесін ойлап шығарған.

Дартмуттік жүйеде тек BASIC – ғана жұмыс істейтін, бірақ ұзаққа созылмай жойылып кетті. Аз уақыттың ішінде Массачусеттік технологиялық институтінің зерттеушілері Bell Labs лабораториясында екінші поколенидағы MULTICS (мультипликациялық ақпараттық және есептеуіш службаны) жүйені өңдей бастады. Кейірек осы лабораторияның зерттеушісі кен Томпсон жаңа қызықтыратын мәселелерді іздеумен шұғылдана бастады.

80-ші жылдардың ортасында UNIX амалдық жүйесі мини-компьютерлерде және инженерлі жұмыс станцияларында қолданылды.

UNIX дербес жүйе. UNIX жүйесі C жоғарғы деңгейдегі тілге жазылғаннан кейін, жаңа машинаға көшіру мәселесі жеңіл процесс бола бастады. Жүйені көшіруде алдымен C тілінен жаңа машина үшін компилятор жазу қажет болды. Содан кейін жаңа машинаның еңгізу-шығару құрылғысы үшін, яғни терминалдар, принтерлер және дискілер үшін құрылғылар драйверін жазу қажет болды.

UNIX амалдық жүйесі PDP – 11 машинадан ауыстырылған алғашқы компьютер – Interdate 8/32 мини-компьютері болады. Мұнда, UNIX жүйесінде бүтін сандар 16-разрядты, адресі 16-битті, және негізгі айнымалыларды сақтау үшін үш регистр болды деген тұжырым пайда болды. Бірақ бұл тұжырымның біреуде ақиқатты болған жоқ. UNIX жүйені ақиқатты етіп жасау үшін біраз еңбектену керек болды.

UNIX жүйесінде желімен жұмыс істеудің барлық жұмыстары екі машинаны қосудан басталды. Interdate – кейін UNIX жүйесі VAX-қа ауыстырылды, ал содан кейін басқа компьютерлерге ауыстырылды.

1984 жылы АҚШ басшылары AT&T корпорациясын бөлгеннен кейін компания қаржылық инвестицияны компьютерлік өндіріске заңды салуды жүзеге асырды. Ұзамай осыдан кейін AT&T компаниясы UNIX жүйесінің бірінші коммерциялық вариантын нарыққа шығарды. Оның нарыққа шығуы аса сәтті болмады, сондықтан бір жылдан кейін ол жаңа версиясына ауыстырылды.

UNIX жүйесінің негізгісі – процесстер болып табылады. UNIX процесстері классикалық процесстерге ұқсайды. Әр процесс бір программаны жүктейді және басқарудың бір ағымын алады. Басқа сөзбен айқанда процессте бір командалар счетчигі бар.

6.3. LUNIX амалдық жүйесі.

Дәріс мазмұны

1. Жүйенің сипаттамасы. Графикалық интерфейс.

2. Мәтіндік редактор N Edit.

LUNIX амалдық жүйесі 1991 жылы құрылып және 1994 жылы Линус Торвальдспен шығарылған Ол басқа амалдық жүйелерден барлық мүмкіндіктерімен ерекшеленеді. LUNIX жазылған ғана амалдық жүйе қызығушымен құрылған (Торвальдс оны дипломдық жұмыс ретінде жазған). Сонымен қатар ол, осы сияқты әлемнің әртүрлі мемлекеттерінің қызығушыларымен күн сайын дамып, толығып және қолдау тауып отарады. Сонымен қатар LUNIX біздің жазылған еркін таралатын амалдық жүйе.

«Еркін таралу» статусы амалдық жүйенің тек бір ғана қасиеті бар, мысалы ашық архитектура. LUNIX ядросы басқа Windows-тан айырмашылығы, «шығу мәтіндері» түрінде таралады және өзгертулер үшін ашық, сол себепті оны әрбір білімді программист оны компьютерге жеңіл және тез қоса алады. Осылайша LUNIX орындалады. Әлемнің миллиондаған программистері өздерінің ескертулері мен қасымшаларын, Линус Торвальд басқаруындағы «Мозговой центр» деп аталатын орталыққа жіберіледі және олардың ішіндегі ең қызықтылары ядроның негізгі версиясына автоматты қосылады.

Осы уақытта LUNIX амалдық жүйесі UNIX-пен біріге бастады, сондықтан оған UNIX – тің программалық қамтамасыздандырулар ауыстырылғаннан кейін LUNIX жүйенің пайдалылығы үлкейді. Сонымен бірге амалдық жүйе LUNIX ОЖ дамуына және кеңейюіне бірнеше адамдар үлесін қосты.

1996 жылы жарыққа 2.0 версиясы шықты. LUNIX жүйесінің бұл версиясы C тілінің – 470000 жолынан, және ассемблер мәтінінің –8000 жолынан тұрды. 2.0 версиясынан кейін қосымша шығарылымдар шыға бастады.

Арнайы LUNIX үшін пайдаланушының екі әртүрлі графикалық интерфейсі жазылды: GNOME және KDE. LUNIX жүйесі тегін үлестірілгенге қарамастан, ұзындығы бойынша Microsoft корпорациясының Windows 2000 амалдық жүйесі үшін лицензиясынан жоғары болды. Пайдаланушылар екілік файлдарды, мәтіндерді еркін көшіреді, үлестіреді. LUNIX және UNIX жүйелерінің ішкі құрламдың деректері, программалары, алгоритмдері өте ұқсас болды. Мысалы, 80 пайызы LUNIX-тің 150 жүйелік шақырулары POSIX, BSD жүйеге ұқсас болды. Сонымен UNIX жүйесін жазудың үлкен бөлігі LUNIX-ке де қолданылады. Жалпы айтқанда екі жүйе де бір біріне ұқсас болып келеді., сол себептен UNIX амалдық жүйенің міндеттерін қарастырайық. UNIX міндеті. UNIX амалдық жүйесі бірнеше процесстерді қолдайтын және бірнеше пайдаланушыларды бір уақытта қолду үшін өндірілген интерактивті жүйе болып табылады. Көп жағдайда программистердің көпшілігі бір жүйені құруда белсіп қызмет етеді, сондықтан UNIX амалдық жүйеде жалпы ақпаратпен бірігіп пайдалануды және программистердің бірігіп жұмыс жасауға мүмкіндік береді.

UNIX утилиталары. Пайдаланушы UNIX интерфейсі тек қабықшадан ғана емес, сонымен бірге стандартты қамтамасыз ету программалардан, яғни утилиталардан тұрады. Жалпы айтқанда бұл программаларды келесідей алты санатқа (категорияға) бөлуге болады:

1. Файлдар мен каталогтарды басқару командалары;
2. Филтрлер;
3. Программаларды өндіру тәсілдері;
4. Мәтіндік процессорлар;
5. Жүйелік басқару;
6. Әртүрлі.

Негізгі түсініктер. UNIX жүйесінің негізгісі – процесстер болып табылады. UNIX процесстері классикалық процесстерге ұқсайды. Әр процесс бір программаны жүктейді және басқарудың бір ағымын алады. Басқа сөзбен айтқанда процессте бір командалар счетчигі бар.

Процестер бір-бірімен байланысып иерархия құруы мүмкін: кейбір жүйеде аталық және балалық процесстер бір-бірімен белгілі байланыста болады, өз кезегінде, балалық процесс жаңа процесстер жасаса, ал оның балалық процесстері арықарай тағы да жаңа процесстер жасаса, онда осылай иерархия құрылады.

Процесс өзінің команда санағышы және ішкі жағдайы бар тәуелсіз объект болса да, оның басқа процесстермен әрекеттесу қажеттігі туады. Мысалы, берілген процесстің шығыс деректері басқа процеске кіріс деректерінің рөлін атқарады.

7 Бөлім. Зертханалық жұмыстар.

7.1 Зертханалық сабақтар 1-9.

Зертханалық жұмыс №1. ms dos-ты іске қосу

Сабақ жоспары

1. Қосымшаны іске қосу

2. Толықэкранды қосу

3. Программаны қосымша жөндеу

Сабақ мақсаты : MS DOS ОЖ-н қатар іске қосды. Толық экранды жұмысқа көшеді.

Зертханалық жұмысқа тапсырма

1. Пуск-Выполнить командасын жіберу

2. Диалогтық терезеде қосымша программаның атауын көрсетіп mem.exe , ОК батырмасын басу керек. MS DOS программасы автоматты түрде іске қосылады.

3. Толықэкранды режим құрал-саймандар тақтасында.

4. нажмите комбинацию клавиш ALT+TAB батырмасын басқанда, жұмыс столы мен басқа қосымшаларға ауысады.

5. Кез-келген батырманы басқанда MS DOS ОЖ- сі туралы ақпарат ала аласыз.

6. MS DOS программасын іске қосуға дайындау. Ол үшін ең алдымен программа құрып алу керек. Жұмыс столының бос жеріне тышқанның оң жағын басып, контекстік менюде Создать-Жарлық батырмасын таңдау керек.

7. Жарлық құру үшін Обзор батырмасын басу керек.

8. Далее батырмасын бас, жарлық атауымен танысып, Готова батырмасын басу керек.

9. Құрылған жарлықта тышқанның сол жақ батырмасын басып, контекстік мненюдағы Свойства командасын таңдау керек.

10. Диалогтық терезеде MS DOS программасын құрып, ОК батырмасын басу керек.

11. Шрифт, Жады, Экран және басқа да диалогтық терезедегі жарлық қасиеттері орналасқан. ОК батырмасын басу керек.

12. Құрылған жарлықты екі рет басу керек.

13. Жұмыс істеп болған соң Windows жүйесі қалай автоматты түрде шамадан тыс жүктеледі.

Зертханалық жұмыс № 2. Тышқанның қасиетімен баптау.

Жұмыс столын безендіруін баптау, Windows 2000 іздеу жұмысы, қоржын және прводникпен жұмыс.

Сабақ мақсаты : Басқару органдарын индивидуальды баптау. Басқару элементін қосу операциясы.

Зертханалық жұмысқа тапсырма

Тышқанның қасиетімен баптау

1. Диалогты терезені ашыңыз Свойства : мыш (Пуск-Настройка-Панель управления - Мышь).

2.Тексеру облысындағы басқару элементіне барып тышқанды екі рет шертіңіз. Тышқанды екі рет шерткенде элементтің ашылатындығына көз жеткізіңіз, ал тышқанды жеке-жеке белгілі бір уақыт интервалы аралығында екі шерткенде ашылмайтынды.

3. Екі рет басу жылдамдығын арттыру мақсатында движокты ең шеткі оң жақ бөлікке жылжытамыз.

Движокты ең шеткі сол жақ бөлікке жылжыта отырып екі бөлік шерту қосарланған екі шертуге интерпретацияланатындығына көз жеткізіңіз.

Өзіңізге қолайлы болатындай движоктың орналасу жағдайын таңдаңыз. Перемещение вкладкасын басыңыз.

Тышқанның сезімталдығын бәсеңдетіңіз движокты ең шеткі сол жақ бөлікке жылжыта отырып. Применить пернесін басыңыз.

Тышқан бағдаршасын экранның сол жақ төменгі бұрышынан оң жақ жоғарғы бұрышына дейін тышқаннан қол үзбей отырып апаруға болатына көз жеткіңіз.

Өзіңізге қолайлы болатындай движоктың орналасу жағдайын таңдаңыз. Применить командасын қолдануды ұмытпаңыз.

Тышқан қасиеті диалог терезесін жабыңыз.

2.Жұмыс столында безендіру баптауы, проводникпен жұмыс, іздеу жүйесі.

Компьютерді қосыңыз, ОЖ–нің жүктелуін тосыңыз, жұмыс столының бос орнында тышқанның оң жақ пернесін шертіңіз.

Контекстік менюден «Свойства» пунктін таңдаңыз - Свойства: Экран диалогтық терезесі ашылады. Фон вкладкасының ашылғандығына көз жеткізіңіз.

Жұмыс столы суреті тізімінен «орман» суретін таңдаңыз, ОК пернесін басыңыз. Жұмыс столының фоны өзгергендігіне көз жеткізіңіз.

1-3-ші пункттерді фон вкладкасындағы суретті өзгерте отырып қайталаңыз. «Созу», «Центрден», «Маңында» тәсілдері экранның безендірілуіне қалай әсер ететіндігін байқаңыз.

1-3 пунктін қайталау, Бұлт суретін таңдап алып, Растянуть әдісімен орнату.

Проводник программасын енгізу.

Проводниктен Windows жүйесін іздеу керек.

Жұмыс столының фондық суретін іздестіру жүйесімен орнату керек.

Орман объектісі табылғанда C: Windows- папкасында іздестірудің қорытындысы көрсетіледі.

Іздестіру жүйесін жабып, проводник терезесіне қайтып оралу.

Жұмыс столында себеттің белгісі көрініп тұруы мүмкін.

Проводник терезесін шерту арқылы жою.

Свойства : Экрандиалогтық терезесінде Фон вкладкасын ашу.

Фондық суреттің тізбегінде Орман суреті жоқ.

Лес объектісін орнату керек.

Фон диалогтық терезесіндегі Своистваны ашып: Жұмыс столының фондық суретінде Лес суреті бар.

Ашық терезелерді жабу.

Зертханалық жұмыс №3. Қосымшаларға автоматты түрде кіру.

Файлдардың типтерінің қасиеттерін өңдеу.

Сабақ мақсаты : Қосымшаларды автоматты түрде шамадан тыс жүктеу.

Файлдардың типтерінің қасиетін редактрлеу.

3. Зертханалық жұмысқа тапсырма

4. *Қосымшаларды автоматты түрде іске қосу.*

1. Компьютерді қосып, ОЖ-нің іске қосылуын тосу керек.

2. Поводник программасын іске қосу.

3. Поводниктің сол бөлігінен C:/. Windows./ негізгі меню/программалар/ автожүктеме құжатын табыңыз. Оны ашып, оң жақ бөлігінен автоматты түрде жүктелген жарлықтар мазмұнын көріңіз. Құжаттың тұрған жерін есте сақтаңыз

/ сол жақ бөлік бойынша автожүктеу.

4. Панельдің сол жағында C:/. Windows құжатын ашыңыз. Оң жағынан Калькулятор (calc/ехе) белгісін табыңыз. Керек уақытында айналдырып көрсетуін пайдаланыңыз. calc/ехе. объектісін іздеуде қиыншылықтар бар болса, объектіні аты бойынша теру командасын басыңыз (Вид – упорядочить значки – по имени).

5. Арнайы тасу командасы бойынша calc/ехе құжатын панельдің оң жағынан сол жағына ауыстырыңыз. Эксперттік түрде сол жақтағы айналдырым автоматты түрде болып жатқанына таситын белгі панельдің шетіне белгіні таситынына көз жеткізіңіз. Тышқанның батырмасын жібермеңіз.

6. Разыскав значок папки / автозагрузка белгісін тауып алғаннан кейін, оған тасу белгісін қойыңыз. Осы жұмыстардың нәтижесі белгіге қойылған жазу түсінің өзгеруі болып табылады. Тышқан батырмасын жібере отырып арнайы тасу мәзірінде Құру (Создать) жарлығын таңдаңыз.

7. Откройте папку / автозагрузка құжатын ашып калькулятор программасының ярлігі пайда болғанына көз жеткізіңіз.

8. ОЖ жұмысын аяқтап, компьютерді өшіріңіз.

9. Компьютерді қосып ОЖ-ң жүктелгенін тосыңыз, калькулятор программасының ярлігі автоматты түрде қосылғанына көз жеткізіңіз.

10. Кез-келген әдіспен / автожүктеу және калькулятор жарлығын жою құжатын ашыңыз.

Файлдар типінің құрылысын редактрлеу.

1. Менің компьютерім (мой компьютер) белгісін тышқанның оң жақ батырмасымен басыңыз. Контексті менюда проводникті ашу пункті бар екеніне көз жеткізіңіз. Екі пункттің де қозғалысын тексеріңіз. Бірінші

жұмыста құжаттар терезесі ашылады, ал екінші терезеде проводник терезесі ашылу керек.

2. Контексті менюде Ашу (Открыть) пункті жартылай жуан шрифпен жазылу,енді осыны алдыңғы фактпен қойып осы жұмыста үндеместікпен істейтініне көз жеткізіңіз (менің компьютерім белгісін екі рет шерткенде). Бұл жаттығудың негізгі мақсаты ол бұл жұмысты өзгерту.

3. Құжат құрылымы диалогтік терезесін ашыңыз (пуск- настройка – свойства папки)

4. Файлдар типі құжатын ашыңыз.

5. Тіркелген тізімдерді айналдыра отырып Құжат(Папка) пункті табыңыз.

6. Өзгерту командалық батырмасын бассаңыз. Құрылымын өзгерту (Изменение свойства) файлдық типі ашылады.

7. Қозғалыс (Действия) тізімінде Ашу және Проводникті ашу құжаттарымен орындалатын екі қозғалыс жазылған. Ашу қозғалысы үндеместік түрінде жасалынған деп саналады, ол жартылай жуан шрифпен жазылған.

11. Проводникті ашу құрылысын белгілеп үндеместің батырмасын басыңыз. компьютерді қосып, ОЖ-ң жүктелгенін тосыңыз, калькулятор программасының жарлығы автоматты түрде қосылғанына көз жеткізіңіз.

12. Кез-келген әдіспен / автожүктеу және калькулятор жарлығын жою құжатын ашыңыз.

Файлдар типінің құрылысын редактрлеу.

8. Менің компьютерім (мой компьютер) белгісін тышқанның оң жақ батырмасымен басыңыз. Контексті менюде проводникті ашу пункті бар екеніне көз жеткізіңіз. Екі пунктің де қозғалысын тексеріңіз. Бірінші жұмыста құжаттар терезесі ашылады, ал екінші терезеде проводник терезесі ашылу керек.

9. Контексті менюде Ашу (Открыть) пункті жартылай жуан шрифпен жазылу,енді осыны алдыңғы фактпен қойып осы жұмыста үндеместікпен істейтініне көз жеткізіңіз (менің компьютерім белгісін екі рет шерткенде). Бұл жаттығудың негізгі мақсаты ол бұл жұмысты өзгерту.

10. Құжат құрылымы диалогтік терезесін ашыңыз (пуск- настройка – свойства папки)

11. Файлдар типі құжатын ашыңыз.

12. Тіркелген тізімдерді айналдыра отырып Құжат(Папка) пункті табыңыз.

13. Өзгерту командалық батырмасын бассаңыз Құрылымын өзгерту (Изменение свойства) файлдық типі ашылады.

14. Қозғалыс (Действия) тізімінде Ашу және Проводникті ашу құжаттарымен орындалатын екі қозғалыс жазылған. Ашу қозғалысы үндеместік түрінде жасалынған деп саналады,ол жартылай жуан шрифпен жазылған.

15. Жұмыс столындағы Менім компьютерімді екі рет шертсе онда проводникте менің компьютерімнің папкасы ашылады.
16. Менің компьютерімдегі папканы ашып, содан прводник ашылуын көру керек.
17. пп 3-9 қайталап, папкілердің қасиеттерін жөндеу керек.
18. Диалогтік терезені жабыңыз.

Зертханалық жұмыс №4. msWindows мәзірі

Жұмыстың мақсаты Мәзірдің барлық түрін үйретіп және оны ажырата білу.

1. MS Windows мәзірін қолдану.

Бас мәзір.

Пуск кнопкасымен шақырамыз.

«Помогает выполнять абсолютно все действия, которые позволительны в Win» іс әрекетін абсолютті түрде орындауын көмектеседі.

Статистикалық және қолданушы бөліктеріне бөлінеді.

Жарлықтар мен пунктардың қызыметін құрайды.

- Программалар жарлықтардың программасын құрайды
- WEB бетін қолданушылар адресіне бейнелеу
- Документтер –15 жарлықтар документтердің соңғы ашылуы.
- Жөндеу – Амалдық жүйесін жөндеу.
- Іздеу – Диалогтық терезені шақырып, файлдарды және папкілерді іздеу.
- анықтама – ОЖ берілгендердің барлық сұрақтарын ақпараттың максимумын құрайды.
- Орындау –жүйелік команданы орындайды.
- Жұмысты тоқтату –Win жұмысын тоқтатады.

тапсырма 1

1. Пуск батырмасын басықанда бас мәзір шығады, осы арқылы бәрін орындауға болады.
2. Мәзірдің астындағы стандарттан калькуляторды табу керек.
3. Жарлықтың калькуляторы арқылы calc.exe программаны аштық. Ашылған терезені жабу керек.
4. Басты мәзір арқылы іздеу командасын тауып алып файлдар мен папкілерді қарастыру.
5. Диалогтық терезедегі барлық вкладкалар мен танысу
6. Терезелердің объектісінде кез-келген файлдарды екі рет шерткенде блокнот программасымен бірге мәтіндік документ ашылады. Ло файлды редактрлейді.
7. Readme.txt жауып, іздеу терезесін кішірейту керек.
8. Басты мәзір арқылы Readme.txt файлды қайта ашады.
9. Readme.txt файла жабу керек.
10. Бас мәзірі арқылы орындау терезесіе ашып, C:/ autoexec.bat файлды табу керек.

11. Бас мәзір арқылы жұмысты аяқтау.
12. Компьютерді қайта іске қосу..
13. пуск/өзгерту/ басқару панелі жолдары арқылы жасаңыз. Компьютердегі кез- келген құрылғыны жөндейтін басқару панелінен жүйелік құжат ашылады, және интерфейс пен ОЖ жұмысы.Мысалға экранды алыңыз.

14. Жұмыс столының уақытша суретін өзгертіңіз.

15. *Контекстің меню*

Объектімен орындауға арналған жұмыстардың тышқанның оң жақ батырмасымен тез орындалуына арналған.

1. Жұмыс столында пиктограмма бойынша контексті менюді шақыруға арналған тышқанның оң жақ батырмасын басыңыз

2. сол жақ батырмамен ашу (открыть) пунктін таңдаңыз. Ашық терезені жаймалаңыз.

3. Контексті менюда менің компьютерім батырмасын шақырыңыз.Объектімен қоса мәтін менюі өзгергеніне көз жеткізіңіз. Жабу пунктін таңдаңыз.

4. Жұмыс столында қоржынды (корзина) шертіңіз, жұмыстарымен танысыңыз.

5. Өзіңізге керек емес атта қоржын (корзина) жарлығын өзгертіңіз.

6. Контексті меню арқылы корзина ярлығын жойыңыз.

7. Жұмыс столының бос орнына контексті менюді шақырып орнатыңыз– объектілер тәртібін аты бойынша жұмыс столының жөнделу бөлігіне шақырыңыз.

8. Экран терезесін жабыңыз.

9. Жүйелік меню

Терезлер өлшемін тақырып жолында ашылған терезеге редактрлеуге шағырады

10. Қоржын (корзина) терезесін ашыңыз

1. Көрсеткішті пиктограмманың сол жақ тақырыптар жолына апарып тышқанның сол жақ батырмасымен басыңыз.Жүйелік меню ашылады.

2. Жүйелік менюді қолдана отырып ашу, жабу, жаймалау командаларын орындаңыз.

3. *Көлденеңдік меню*

Берілген объектілер бойынша берілген терезеде жұмыстарды орындауға мүмкіншілік береді.

11. Корзина терезесін ашыңыз

12. Объектіні терезеде белгілемеңіз. Меню батырмаларымен таңдай отырып көлденең менюдің пунктерімен танысыңыз.

13. вид/спис Readme.txt ок орындаңыз

1. файл және түзету (правка) пунктінде өзіңізге керекті жол мен вдавлен/отключены шрифті таңдаңыз

14. қоржындағы өзіңізге керек емес жарлық пиктограмма з құжаттын белгілеңіз. файл и правка пунктінде қосылған жұмыстардың мөлшері көбейеді.

15. файл/жою (файл/удалить) командасын орындаңыз.

Басты менюмен жұмыс:

- Справканы шақырып демонстрациялық ресурстарда тақырыптар көрсеткішін шақырыңыз.Берілгенді оқып орындаңыз.

- Жұмыс столында қоржынды тышқанның оң жақ батырмасымен ұстап алып пускке апарыңыз. Жарлық құрыңыз.Оның артықтылығына көз жеткізуді пуск батырмасы арқылы тексеріңіз.

Терезелермен жұмыс:

- контексті меню арқылы менің компьютерім ярлігін құрыңыз,оны қолданыс компьютеріне ауыстырыңыз, және жойыңыз.

- Контексті меню арқылы менің компьютерім құжатын ашыңыз.

- Жүйелік менюді қолдана отырып оның өлшемін өзгертіңіз.

Зертханалық жұмыс № 5..Windows ОЖ-ң файлдық құрылымы.

Жұмыс мақсаты: ПК-ға мәліметтерді сақтаудың әдістері,және ОЗУ-ң ішкі тасымалындарында да.

Windows ОЖ құжаттары мен қолданыстары.

Құжаттарын қолдану

1. Жұмыс столында My first box қолданушылар құжатын құрыңыз. Ол үшін жұмыс столының бос орнына контексті менюді,оның ішінде құжат құру командасын шақырыңыз.Бәрден тышқанды босатып пернетақтамен басқа бос орынға My first box мәтінін теріңіз. Ол шығару (ввод) батырмасын басумен пайда болады.

2. контексті меню арқылы осы құжатқа арналған құрылымын ашыңыз. Диалогтік терезеде осы құжаттың құрамын және құрылған күнін ,орналасу жерін оқыңыз.Бұл терезеде атрибудтар жұмысы болуы мүмкін,мысалы, тек қана оқуға арналған (объектіні редактрлеумен шектеледі. Кері қайтаруды басыңыз (отмена)).

3. жазу бойынша емес, пиктограмма бойынша құжатты ашыңыз.Терезе интерфейсімен танысыңыз.

4. Myfirstbox салынған құжатын құрыңыз . Көлденең немесе контексті менюді қолданыңыз.Атын қолданыңыз.

5. Myfirstbox құжатын көшіріп терезені кішірейтіңіз. Myfirstboxқұжатының белгісін жұмыс столына тасыңыз. Оны көшіріп алыңыз.

6. Құжатты жойып жіберіңіз.Қоржынды тазартыңыз.

7. Myfirstboxқұжатын кезекші атына ауыстырыңыз (файл – переименовать).

8. Кезекші құжатын жұмыс столына көшіріңіз (переместить)

9. жұмыс столында кезекшіні тышқан көмегімен тасып, пернетақтамен Shiftбатырмасын басу арқылы қоржын пиктограммасына әкеліңіз. Бұл құжатты қоржынға апарумен жойып жібердіңіз.

Файлдармен жұмыс

1. Құжатта My first box файлы құр, типі мәтіндік құжат (файл-создать – текстовый документ). Бұл құжатқа визитка.txt атын беріңіз, txt –құжаттың кеңейтілуі.

2. Осы құжатты ашыңыз оған блокнот программасы қосылды

3. Осы берілімге өзің туралы үш жолдан тұратын мағлұмат енгізіп блокнотты жабыңыз.

4. Атын сол қалпында қалтырып файл атын өзгертіңіз, кеңейтілуі DOC болу керек, файлды ашқанда өзгертілген кеңейтілуден кейін MSWord программасы жүктеледі.

5. Контексті меню арқылы визитка файлы жұмыс столынан көшіріп алыңыз.

6. \визитка. DOC. \ жұмыс столынан жойыңыз-оны құжаттар мен файлдар бөліп тұрады.

Жады аймағында уақытша тасу және көшіру жұмыстары келесі ауыстырылымға дейін объект сақтайды.

1. визитка. DOC файлы белгілеңіз, ары қарай көлденең меню арқылы жөндеу/көшіру (правка \ копировать) командасы арқылы ауыстыру буферін таңдап алыңыз.

2. объектіні жұмыс столына қойыңыз.

3. жұмысты қайталаңыз

4. ауыстыру буфері тек қана көшіру емес объектіні тасуы да мүмкін. Ол үшін кесіп алу батырмасын басыңыз.

5. өз файлыңызды жойыңыз

Проводник программасы

1. Негізгі меню арқылы проводник құжатын ашыңыз.

2. программа интерфейсін оқыңыз: сол жақ панель құжаттан, оң жақ панель ашылған құжаттардан тұрады

3. құжаттарды жаймалаңыз. (-)

4. с арқылы (+) белгісін басыңыз

5. түр (вид) пунктін ашып саймандар тақтасында жалауша тұрғанына көз жеткізіңіз, егер жоқ болса осы пунктін шертіңіз. Батырмалар жұмысын дәптеріңізге жазып алыңыз.

6. Дүзету-қою (Правка - Вставить) файлы көшіріп алыңыз.

7. тұрған файлды ауыстырыңыз.

8. Проводник программасында Myfirstbox сурет файлы құрыңыз сурет.bmp файл \ создать \ точечный рисунок пункттері арқылы

9. Визитка аты бойынша барлық файлдарға арналған іздеу салыңыз. (проводникте сервис/поиск/файлы файлы шақырыңыз және іздеу құжатында іздеуіне шарт қойыңыз : аты – визитка, құжат – менің компьютерім).

10. барлық құжаттарды жойыңыз
11. проводник программасын жабыңыз

Өздік жұмыс

1. Проводник арқылы жұмыс столында құжаттар бұтағын құрыңыз: жұмыс столы\әдебиеттер\классика
2. Ақындар файлдарын проводникте іздеп классика құжатын көшіріп алыңыз
3. классика құжаттық ақындармен аталған файл құрыңыз. Жұмыс столында тасу және контексті меню арқылы ақындар құжатының ярлігін құрыңыз

Зертханалық жұмыс №6. Жарлықтармен жұмыс. Жарлықтар, бас менюді түзету.

Жұмыс мақсаты: жарлық түсінігін кеңейту. Оны құрып қолдануды үйрену.

Жарлықты қолдану.

1. Бас менюдің ішінен программалар –тұрақты- paint. Paint ашыңыз.– бұл жарлық Ms paint.exe программасы үшін.
2. оған әртүрлі әдістер арқылы көз жеткізуге болады , мысалы, пуск\іздеу\ файлдар және құжаттар, бұнда маскіге ат беріледі * paint* (* кез-келген символға кез- келген сан дегенде білдіреді).
3. программаның көмегімен проводник программасы арқылы маршрут бойынша программаны ашады.
4. paint терезесінің екеуін де жабыңыз
5. сол маскамен жарлықтарға іздеу салыңыздар, қосымша жарлық-типін беріңіз (тізім алфавит бойынша)
6. көрсеткішті бөлгішке қою арқылы өз терезеңде маршруттар санын көбейту үшін құжаттар өрісін жылжытыңыз.Құжаттар өрісін кішірейту керек,өйткені жарлықтың өлшемі ылғи да 1 килобайт және күнді өзгертуді салыстыру.
7. Екі рет табу жарлығын екі рет басыңыз *- жарлық құру пиктограммасы ашылады
8. Жарлықты бас менюде ауыстыруға болады. Қоржын пиктограммасын алып пуск бойынша жүргізіңіз, менюдегі жарлықтар санын тексеріңіз
9. Бас менюді ашыңыз, ол үшін пускты тышқанның оң жақ батырмасымен таңдап алып ашу командасын басыңыз.
10. Ең бірінші құжат программасына , одан кейін тұрақтылар программасына өтіңіз. Ауыстыру буферіне WordPad ярлығын көшіріңіз (Түзету\Көшіру), тышқанның оң жақ батырмасымен жұмыс столын шеру арқылы қоюды таңдаңыз.

Тұрақтылыр терезесін жабыңыз.

Жарлық құрудың универсалды тәсілі.

Жарлық құрылатын файлдың тура орыны (пуск-іздеу – файлдар және құжаттар)

- керекті құжатқа жарлық құру (создать \жарлық) файлын шақыру
- обзор көрсеткіші арқылы файладың жолын көрсету
- Жарлық дайын атын беру.
- Белгіні ауыстыру (менюді шақыру/ құрылымы/ вкладка жарлық /белгіні өзгерту батырмасы / белгіні таңдау / ок).

Контекстное меню арқылы жарлық құру.

- Жарлық құрылатын файлды белгілеп алу.
- Тышқанның оң жақ батырмасымен жарлық қойылатын жерді белгілеп алу.
- Батырманы жіберу.
- Жарлық құруды таңдап алу.

Басты менюден көшіріп алу.

- Пусктің оң жақ батырмасы арқылы ашу.
- Керекті құжатқа кіру.
- Айырбастау буфері арқылы көшіріп алу.

Бас менюді редактрлеу

- Пуск батырмасына көшіру: жарлық қажет ететін файл белгісін таңдап алу.
- Пускке апару.

Құрылым терезесінде берілімдер панелі

пуск/ жөндеу/жұмыс панелы және пуск./ вкладка настройка меню

Контексті меню:

- пуск
- объектіні таңдау
- контексті меню және керекті пункті таңдап алу

Зертханалық жұмыс №7. Процестерді басқару.

Жұмыс мақсаты: процесстерді басқаруда жаңа нәтижелер алу және командалық құжаттармен өздік жұмыс.

Процестермен жұмыс істеуге арналған POSIX командасы (барлық амалдық жүйелерде болу керек)

at – программаны белгілі уақытта жібереді

crontab – кестені жұмысқа жіберетін файлдар

kill - PID процессі арқылы процестің орындалуын тоқтату

nice –процесті жүктеу кезінде приоритет сұрайды

renice – жұмыс істеу процессінің приоритетін ауыстырады

ps – жұмыс істейтін процесстердің хабарламасын шығарады

fg – түстік режимнен процесті аудару

bg – түстік процесс орындалуын жалғастыру, егер ол <Ctrl+Z>

батырмасы арқылы тоқтатылған болса

Процесстермен жұмыс істеуге арналған LINUX командалары

at – программаны белгілі бір уақыт бойынша жүктейді
atq – at командасы қойған тізімге жұмыс тізімін шығарады
atrm – at командасының тізімі бойынша жұмысты жою
/etc/crontab – кестеден тұратын файлдың жұмысын шығару
kill - PID процесі бойынша процесті тоқтату
killall – процессаты бойынша процесті тоқтату
nice –процесті жүктеу кезінде приоритет сұрайды
renice – жұмыс істеу процессінің приоритетін ауыстырады
ps – жұмыс істейтін процесстердің хабарламасын шығарады
top – процесстерде динамикалық мағлұматтарды шығарады
fg - процесті түстік режимнен шығару
bg – <Ctrl+Z> батырмасы арқылы тоқтатылған болса, түстік процесті орындауды жалғастырады
iircs – процесстердің бірдей жұмыс істеуі (бөлінетін жады, семафорлар, хабар)

Толық мағлұмат алу үшін help қолдануға болады (мысалы: ps -- help), немесе құжаттарға (мысалы: man ps, шығу үшін q-ды басасыз).

Түстік процесті жүктеу былай орындалады:

ps -x &

Жүйені жүктеуде керекті процесстер түстік режимге жүктеледі, оларды "демонами" деп атайды. Олар/etc/rc.d/init.d/каталогында болады.

Пернетақтаның кейбір комбинациялары:

<Ctrl+Z> - жұмыстың орындалуын уақытша тоқтату

<Ctrl+C> - жұмыстың орындалуын тоқтату

Каналдардың көмегімен процесстерді байланыстыру. Бірнеше командаларды демалыстық ағыммен келесі программаға жіберілуі, "|" демалыстық ағымды бірінші программдан екінші программаға жіберудің белгісі.

ps -ax | more

ps –ax командасы жіберіледі, және more демалыстың ағымын программа орындауды жібереді.

Енгізу/шығаруды кері бұру. Демалыс ағымы бар команданы файлға жіберу

ps -ax > test.txt

ps -ax > test.txt –файлдың өзіне қосу

Командалар тобы

command-1;command-2;command-3

{command-1;command-2} > test.txt

Процесстермен жұмыс істеуге арналған Windows командалары

Процессор туралы мағлұматтардың көбін жұмыс диспетчеренен (диспетчерзадач) білуге болады.

at – берілген уақытта программаны жүктеу
Schtasks –кесте бойынша командалардың орындалуын жөндейді
Start – жеке программаны немесе команданы жеке терезеде көрсетеді

Taskkill – процесті аяқтайды

Tasklist –Жұмыс істейтін процесстер туралы мағлұмат шығарады
Көбірек мағлұмат алу үшін сұрау орталығы мен көмек көрсетуіне немесе help командасына көңіл аударуға болады (мысалы: help at)
command.com - MS-DOS командалық оболочкасын жүктеу
cmd.exe – Windows командалық оболочкасын жүктеу

Зертханалық жұмыс №8. Жұмыс диспечерімен жұмыс.

Жұмыс мақсаты: Жұмыс диспечерімен жұмыс істеуді үйрену.

1. Windows-ты жүктеңіз.
2. Жоғарғы командаларға қарап қандай параметр не үшін керек екенін біліп сұрынымын құрыңыз (орыс тілінде).
3. Мына командалармен жұмыс істеңіз: жүктеу, уақыт бойынша жүктеу, приоритеттерді ауыстыра отырып, процесті жою арқылы.
4. Жұмыс диспетчерімен қалай жұмыс істеу керек және баған туралы мәлімет нені көрсетеді екенін көріңіз (жады мәліметінен басқа).
5. Нені иемдену керек: (командалық бағанмен):
6. белгілі бір уақыт бойынша процеске жұмысты жүктеу және белгілі проритетпен осы тізімді көрсету және тізімнен жойып тастау.- жеке программаны немесе команданы жеке терезеге жүктейді - процесті аяқтау-жұмыс істейтін процесстер мен ағымдар мәліметтін шығару (және ұғу), (енгізу/шығару жадысын есептемегенде).

Зертханалық жұмысқа қосылатын: процесстерді басқаруға арналған командалар сұранысы және осы командалармен жұмыс істей алуы.

Зертханалық жұмыс №9. Іnux аж –де жұмыс.

Linux-ті жүктеңіз.

1. Жоғарғы командаларға қарап қандай параметр не үшін керек екенін біліп сұрынымын құрыңыз (орыс тілінде).
 2. Мына командалармен жұмыс істеңіз:жүктеу,уақыт бойынша жүктеу, приоритеттерді ауыстыра отырып, процесті жою арқылы.
 3. Нені иемдену керек: белгілі бір уақыт бойынша процеске жұмысты жүктеу және белгілі проритетпен,осы тізімді көрсету және тізімнен жойып тастау.
- /etc/crontab файлын конфигурация жасау-ld және аты бойынша процстті тоқтату белгісін беру- нақты приоритет бойынша процесті жүктеу
- нақты процесс бойынша приоритеттерді өзгерту
 - жұмыс істейтін процесстер бойынша мәліметтерді шығарады
 - түстік режимде процесті жүктеу және шығару

- басқа программада демалыс ағымын жіберу
- демалыс ағымының процесін файлға жазу

7.2 Зертханалық сабақтар 10-15

Зертханалық жұмыс №10-15. Linux амалдық жүйесімен жұмыс .
Netstat утилитасы.

Жұмыс мақсаты: netstat утилиті арқылы желілік IP- торын зерттеу.

Зертханалық жұмысқа тапсырма

1.1. netstat утилиті арқылы pds.sut.ru серверінде ашылып тұрған тізімдермен байланыс алу. ESTABLISHED режимінде болатын байланысқа коментария жасау (ішкі немесе сыртқы интерфейс, қандай байланыспен, қай протоколға).

1.2. Жүрген жолының кестесін алу. Қандай интерфейстер арқылы қандай торлармен байланыс жүруде, шлюза аты, желілік торлардың маскасы қандай екенін көрсету.

1.3. Желілік интерфейстер статистикасын алу. Ішкі статистикалық мәліметтер үшін (eth0) және сыртқы (eth1) жергілікті торлар үшін графигін құру, қойылған жалауларың мәнін түсіндіру. Желімен жұмыс істеуге арналған қателер санын салыстыру, жергілікті тор жұмысының нәтижесін шығару.

1.4. Желілік интерфейсстің статистикасын алу. Әрбір протокол бойынша жұмыстарға анализ жасау. ICMP протоколына кіріс және шығыс гистограммасына байланысты график құру.

2. Зертханалық жұмысқа әдістемелік көрсетулер

Netstat утилитасы жергілікті тор және TCP/IP қолданысы туралы мәліметтердә шығарады. Дәл осыған көбнесе TCP/IP торының жөнделмеуіне себепті тез іздеу үшін администраторлар көңіл аударады. Демалыс мәліметінің мазмұны және формасы амалдық жүйеге байланысты, бірақ келесі мәлімдер шығарады: байланыс тізімі, желілік интерфейсстің статистикасы, берілгендер буфері бойынша мәліметтер, жүрген іздері бойынша кесте мазмұны, протокол жұмысы бойынша статистика. Берілген мәліметтердің мінездемесін жолдық командалар операциясы бойынша таңдауға болады.

Мынаған көңіл бөліңіз:

Утилита параметрлерін экранға шығарғанда мына командаларды орындаңыз | more шеткілік шығару үшін. (4-кесте)

4-кесте. Netstat утилита функциялары

Кілт	Функциялар
-r route	Жүрген іздерінің кестесін шығару
-i interfaces	Желілік интерфейсстің статистикасын шығару
-s statistics	Мәліметтерді жіберу статистикасын шығару

	(SNMP протоколы бойынша)
-n numeric	Сандық түрде порттардың аты
-N symbolic	Символдық түрде порттардың аты
-l listening	Күту режиміндегі порттардың жағдайын шығару
-a all	Барлық порттардың жағдайын шығару
Порттар арқылы белсенді қосылу:	
-st	TCP
-u	UDP
-W	RAW
-X	UNIX

2.1 Байланыстар тізімі.

Netstat командасы активті және пасивті жағдайдағы порттарды көрсету кілттер жинағынан тұрады. Осылайша берілген компьютерде барлық серверлік тізімдерді алуға болады. Мәліметтер баған бойынша беріледі. Оның біріншісі болып протокол, одан кейін кезектік қабылдау өлшемі және байланыстар бойынша берілген машинаға (басқа жағында кезектік өлшемі басқадай болуы мүмкін), жергілікті және жойылған мекені және қазіргі байланыс күйі құрылған.

Мысалы:

```
st1@pds:~ > netstat -ta
```

```
Proto Recv-Q Send-Q Local Address      Foreign Address    State
tcp    0      2 pds.sut.ru:telnet  gerasim.pds.sut.ru:1288 ESTABLISHED
tcp    1      0 pds.sut.ru:4550    pds.sut.ru:3128    CLOSE_WAIT
tcp    1      0 pds.sut.ru:4548    pds.sut.ru:3128    CLOSE_WAIT
tcp    0      0 gw.pds.sut.:netbios-ssn marya.pds.sut.ru:1027 ESTABLISHED
tcp    0      0 gw.pds.sut.:netbios-ssn yanko.pds.sut.ru:1104 ESTABLISHED
tcp    0      0 gw.pds.sut.:netbios-ssn mumu.pds.sut.ru:1065 ESTABLISHED
tcp    0      0 *:6000             *.*                LISTEN
tcp    0      0 *:3128             *.*                LISTEN
tcp    0      0 *:53333            *.*                LISTEN
tcp    0      0 *:389              *.*                LISTEN
tcp    0      0 localhost:1032     localhost:1033     ESTABLISHED
tcp    0      0 *:netbios-ssn     *.*                LISTEN
tcp    0      0 *:smtp             *.*                LISTEN
tcp    0      0 *:imap2            *.*                LISTEN
tcp    0      0 *:pop3             *.*                LISTEN
tcp    0      0 *:login            *.*                LISTEN
tcp    0      0 *:shell            *.*                LISTEN
tcp    0      0 *:8000             *.*                LISTEN
```

```

tcp 0 0 *:telnet      *.*          LISTEN
tcp 0 0 *:ftp         *.*          LISTEN
tcp 0 0 *:time       *.*          LISTEN
tcp 0 0 *:www        *.*          LISTEN
tcp 0 0 *:2049       *.*          LISTEN
tcp 0 0 *:832        *.*          LISTEN

```

--More--

Мысалда көрсетілгендей серверлердің көбі (LISTEN) сұраныс режимінде орналасқан. Бірінші жолда (ESTABLISHED) байланысы telnet тышқан gerasim.pds.sut.ru. арқылы орналасқан. CLOSE_WAIT түрі байланыс үзілді деген мағынаны береді, бірақ LISTEN түріне қосылуы әлі болған жоқ. TIME_WAIT-бұл байланыс үзілісті тосуда деген мағынаны білдіреді. Егер байланыс SYN_SENT түрінде болса, онда жоқ сервермен байланыс орнатуға тырысатын процестің басым болуын көрсетеді. Байланыстың қалып-күйі тек TCP протоколы үшін ғана маңызды. UDP протоколы байланысты орнату үшін тексерілмейді.

2.2. Маршруттық кестенің байланысы.

Әрбір машинаның желімен байланысы желілік интерфейс деп аталады. Бір интерфейсін бар машина мәліметтерді тек қана бір интерфейс бойынша қабылдауы және басқаға беруі мүмкін, осылайша мәліметтерді торлар арасында жіберіп отыруы мүмкін. Бұл функция маршрутизация деп, ал осы процесті орындайтын машина-шлюз деп аталады.

Маршрутизацияның берілгені ядроның бір кестесінде сақталады. Белгілі бір мекенжай бойынша белгілі бір пакетті жіберу үшін қолданылады. Егер ондай маршрут жоқ болса, қателік туралы хабар келеді. netstat-г командасы маршрутизацияның кестесін көрсетіп тұрады. Кездесу пункті машинаның аты бойынша немесе IP-мекенжайы бойынша көрсетілуі мүмкін. Флагтар маршруттың бағасын береді.

Мысалы:

```

st1@pds:~ > netstat -r
Kernel IP routing table
Destination Gateway Genmask Flags Ifac
pds.sut.ru * 255.255.255.255 UH eth1
195.19.219.120 * 255.255.255.248 U eth0
195.19.219.128 * 255.255.255.192 U eth1
192.168.1.0 * 255.255.255.0 U eth0
195.19.221.0 lgw.ccs.sut.ru 255.255.255.0 UG eth1
193.125.0.0 lgw.ccs.sut.ru 255.255.0.0 UG eth1
Loopback U lo
Default lgw . ccs . sut . ru 0.0.0.0 UG eth 1
(5 кесте)

```

5-кесте. Netstat утилиты бағандары

Баған аты	шифрі
Gateway	қолданылатын шлюздардың аттары
Genmask	маска , қолданылатын елестетуге арналған мекенжай жалпы бөлімдері , сапар желісін лайықтың сона табы
Flags	жалаулар , сапар желісі суреттеуші :Gсапар желісі шлюзді қолданады
	U интерфейсі белсенді, мәліметтерді жіберу үшін қолдануға болады
	H мәліметтерді тек бір ғана түйін бойынша жіберуге болады
	D Жазу ICMP протоколын кері қайтару үшін құрылған
	M жазу ICMP протоколын кері қайтару үшін модификацияланған
Iface	интерфейс, пакеттерді тапсыру үшін арналған

2.3. Желілік интерфейсдер статистикасы.

Кілт қолдануда- экранға netstat командасын і статистикалық табы арқылы барлық қолданылатын интерфейсдерді толық шығарылған болады. Осыдан тормен байланыс жөнделген бе екенін анықтауға болады. Мысал :

```
St 1@ pds :~>netstat - i
Kernel Interface table
----- OVR Flg
Eth 01000084490401701454454500 BRU
Al
Eth 1150005908440704344385900 BRU
Lo 392404575400045754000 LRU
```

Қалыпты жұмыс істеуші торларда конфликт (RX-OVR, TX-OVR) пакеттердің санынан 3% шамадан асыруы тиіс, ал басқа қателер 0,5% пакеттердің санынан аспау керек .

2.4. Мәліметтерді тапсыру статистикасы. netstat-командасы желілік программаларға санаудың мазмұнын береді . Шығатын мәліметтерде әртүрлі потоктарға байланысты бөлімдер бар: IP, ICMP, TCP, UDP. Осыдан қабылданған пакеттің қателіктерін тексеруге болады.

7.3 Емтиханға дайындық сұрақтары

1. Амалдық жүйе дегеніміз не?
2. ОЖ даму тарихы.
3. ДЭЕМ – нің жұмыс режимі.
4. Амалдық жүйелердің қандай түрлерін білесіздер?
5. Дербес компьютердің негізгі компоненттері мен қосымша құрылғыларын атап шығыңы
6. ДЭЕМ –де қандай принтерлер қолданылады?
7. Дербес компьютердің қосымша құрылғыларын атап шығыңыз?
8. ДЭЕМ – де қандай принтерлер қолданылады.
9. Қандай бейне адаптерлерін білесіздер, дисплей мен бейне адаптерінің айырмашылығы қандай?.
10. ДК-де қандай дискілермен жұмыс істеуге болады?
11. Модем дегеніміз не және ол не үшін қажет?
12. Дербес компьютердің негізгі компоненттері мен қосымша құрылғыларын атап шығыңыз?
13. ДЭЕМ –де қандай принтерлер қолданылады?
14. Процессор дегеніміз не?
15. Виртуалды жад дегеніміз не?
16. Процессор жұмысының принциптері.
17. Интерфейс дегеніміз не?
18. Интерфейстің түрлерін атаңыз.
19. Стандартты интерфейс дегеніміз не?
20. Пайдаланушының графикалық интерфейсі деген не?
21. Графикалық интерфейстің негізгі элементтерін атаңыз.
22. MS-DOS амалдық жүйесі.
23. Файл дегеніміз не?
24. Ішкі файл дегеніміз не?
25. Каталог дегеніміз не?
26. Дискіге файлды көшірудің қанша тәсілі және файлды сақтаудың қандай жолдары бар?
27. MS-DOS амалдық жүйесінің негізгі атқаратын қызметі қандай?
28. Windows 95/98/XP жүйесі
29. Windows 7 жүйесі.
30. ДЭЕМ – нің жұмыс режимі.
31. Амалдық жүйелердің қандай түрлерін білесіздер?
32. Windows 95/98/XP жүйесіндегі компьютерді өшіру және іске қосу әрекеттерін атап беріңіз.
33. OS/360/370/375 амалдық жүйесінің қандай мүмкіндіктері бар?
34. ДЭЕМ – нің жұмыс режимі.
35. Амалдық жүйелердің қандай түрлерін білесіздер?
36. UNIX амалдық жүйесі.
37. LINUX амалдық жүйесі

38. UNIX және LINUX амалдық жүйелерінің ұқсастығы мен айырмашылығы.

39. ОЖ даму тарихы.

40. MS-DOS амалдық жүйесі және NORTON COMMANDER арасындағы айырмашылығы.

41. Амалдық жүйелердің түрлерін

42. RSX амалдық жүйесіндегі компьютерді өшіру және іске қосу әрекеттерін атап беріңіз.

43. Синхронизация есебі. Процестер арасындағы қарым-қатынас.

44. LINUX амалдық жүйесінің қандай мүмкіндіктері бар?

45. Процесті жасау. Процестің аяқталуы. Процестер иерархиясы.

46. ДЭЕМ – нің жұмыс режимі.

47. Амалдық жүйелердің қандай түрлерін білесіздер?

48. LINUX амалдық жүйесіндегі компьютерді өшіру және іске қосу әрекеттерін атап беріңіз.

49. . Linux (EXT2; EXT3; RFS; JFS; XFS) файлдық жүйелері, қызметі, ерекшелігі.

50. Монитор дегеніміз не?

51. Қандай монитор монохромды болып есептеледі.

52. Процестер арасындағы қарым-қатынас, оларды басқару

53 CD-ROM (ISO 9660, UDF) файлдық жүйесі, параметрлері

54 Жадыны басқару. Беттік ұйымдастыру.

55. Процестерді өзара бекіту

56. Тегеріштер және таймерлер ұғымы, қызметі, параметрлері.

57. Құрылғылар және енгізу-шығаруды программалық қамтамасыз ету.

58. Амалдық жүйелердің (ОЖ) концептуалды негіздері. Тарихы. Қызметі.

59. Процестер анықтамасы және классификациясы.

60. Жүйелік шақырулар. Амалдық жүйелер құрылымы.

61. UNIX V7; BSD файлдық жүйесі, қызметі, ерекшелігі.

62. Ресурс ұғымы. Түрлері, ресурстарды тарату әдістері

63. Беттерді орналастыру алгоритмі.

64. Ағындар ұғымы. Қолдану ерекшелігі. Си ++ тіліндегі ағындармен жұмыс істеуге арналған функциялар, операторлар.

65. Процестерді жоспарлаудың негізгі түсініктері

66. Амалдық жүйелер механизмдері.

67. ОЖ-ресурстарды басқару жүйесі ретінде.

68. Файлдық жүйе. Файлдар. Каталогтар. Файлдық жүйелерді тарату

7.4. Тест тапсырмалары.

1. Windows жүйесіндегі MS(PC)-DOS амалдық жүйесі

Жұмыс істеу негіздерінің басталу уақыты

- a) 1981
- b) 1900
- c) 1800
- d) 1780
- e) 1692

2. Экранның төртбұрыштық қоршаулы аймағы.

- a) Терезе
- b) Стол
- c) Ворд
- d) Пиктограмма
- e) драйвер

3. Ең алғашқы MS-DOS амалдық жүйесі компьютердің тек неше Кб жедел жадын ғана (RAM) пайдалана алатын болған

- a) 640
- b) 540
- c) 440
- d) 340
- e) 240

4. Терезе деген не?

- a) экранның төртбұрышты қоршаулы аймағы.
- b) онда әртүрлі программалар орындалады
- c) Кез-келген мәліметтер өңделіп түзетеді
- d) бұл терезенің периметр бойынша өтетін тік және көлденең сызықтар
- e) бұл терезеде негізгі қолданбалы программалар жұмысы атқарылады.

5. Терезе шекаралары деген не

- a) экранның төртбұрышты қоршаулы аймағы.
- b) онда әртүрлі программалар орындалады
- c) кез-келген мәліметтер өңделіп түзетіледі
- d) бұл терезенің периметр бойынша өтетін тік және көлденең сызықтары
- e) бұл терезеде негізгі қолданбалы программалар жұмысы атқарылады.

6. Қолданбалы программа терезесі деген не

- a) экранның төртбұрышты қоршаулы аймағы.
- b) Онда әртүрлі программалар орындалады
- c) Кез-келген мәліметтер өңделіп түзетіледі
- d) бұл терезенің периметр бойынша өтетін тік және көлденең сызықтар

е) бұл терезеде негізгі қолданбалы программалар жұмысы атқарылады.

7. Кез-келген терезенің белгілі бір шектеулі мөлшері болады, сондықтан информация көп болса, ол терезеге симай қалады. Мұндай жағдайда терезенің оң жақ шетінде (төменгі шетінде) айналу сызықтары деп аталатын не пайда болады.

- a) Элемент
- b) Терезе
- c) Драйвер
- d) фото
- e) жарлық

8. Қазіргі кезде Windows 8 –ң неше түрлі нұсқада таралған

- a) 3
- b) 4
- c) 5
- d) 6
- e) 7

9. Жеке компьютерлерге арналған Амалдық жүйеге мысал келтір

- a) Microsoft Windows, Linux, Mac OS (Darwin) және Unix
- b) Офис
- c) Индрайвер
- d) Деректер базасы
- e) Unix.

10. Microsoft Windows ОЖ қай жылы ойлап тапты

- a) 1985
- b) 1999
- c) 2000
- d) 2012
- e) 1988

11. Microsoft Windows ОЖ 1985 жылы қай күні жарыққа шықты

- a) 20 қараша
- b) 22 ақпан
- c) 22 май
- d) 22 март
- e) 22 июнь

12. Windows 2000 ОЖ-і қай жылы жарыққа шықты

- a) 1994
- b) 1995
- c) 1996
- d) 1997
- e) 1999

13. Windows 2000 Microsoft корпорациясының Windows NT отбасындағы неше биттік процесстермен жұмыс жасайтын амалдық жүйесі.

- a) 32
- b) 30
- c) 28
- d) 24
- e) 22

14. Тұңғыш бета-нұсқасы жұртшылыққа 1997 жылдың қай күні көрсетілді.

- a) 27 қыркүйек
- b) 27 март
- c) 28 июнь
- d) 25 қараша
- e) 25 мамыр

15. Тұңғыш бета-нұсқасы жұртшылыққа қай жылдың 27 қыркүйегінде көрсетілді.

- a) 1997
- b) 1998
- c) 1999
- d) 2000
- e) 2001

16. Windows-8 амалдық жүйе әлемнің неше тіліне аударылған.

- a) 37
- b) 38
- c) 39
- d) 40
- e) 41

17. Windows-8 амалдық жүйе әлемнің қанша елінде сатылымға түседі

- a) 140
- b) 130
- c) 120
- d) 110
- e) 200

18. Windows Xp ОЖ қай жылы жарыққа шықты

- a) 2001
- b) 2000
- c) 2002
- d) 2003
- e) 2004

19. Windows Xp ОЖ 2001 жылдың қай күні жарыққа шықты

- a) 25 қараша
- b) 22 мамыр
- c) 25 март
- d) 31 қазан
- e) 11 қараша

20. Амалдық жүйелер деген не
- a) Компьютердің құрылғыларын басқаратын, пайдаланушы мен компьютер арасында байланыс орнататын жүйе болып табылады
 - b) Онда әртүрлі программалар орындалады
 - c) Кез-келген мәліметтер өңделіп түзетеді
 - d) бұл терезенің периметр бойынша өтетін тік және көлденең сызықтар
 - e) бұл терезеде негізгі қолданбалы программалар жұмысы атқарылады
21. Компьютердің құрылғыларын басқаратын, пайдаланушы мен компьютер арасында байланыс орнататын жүйені не деп атаймыз
- a) Microsoft Windows, Linux, Mac OS (Darwin) және Unix
 - b) Амалдық жүйе
 - c) Индрайвер
 - d) Деректер базасы
 - e) Unix.
22. 2009 жылы қай амалдық жүйе шықты
- a) Windows 7
 - b) Windows 8
 - c) Windows 8.1
 - d) Windows 10
 - e) Windows 9
23. Windows 7 амалдық жүйесі қай жылы шықты
- a) 1997
 - b) 2007
 - c) 2003
 - d) 2005
 - e) 2009
24. 2009 жылдың қазан айында нарықтың 18% құраған амалдық жүйе
- a) Windows vista
 - b) Windows 8
 - c) Windows 8.1
 - d) Windows 10
 - e) Windows 9
25. 2007 жылда нарықта пайда болған амалдық жүйе
- a) Windows vista
 - b) Windows 8
 - c) Windows 8.1
 - d) Windows 10
 - e) Windows 9
26. Индекстердің неше түрі бар
- a) 2
 - b) 3
 - c) 5

d) 6

e) 1

27. Жүйелік дискета не үшін керек:

a) Амалдық жүйені жүк

b) Компьютердің барлық компоненттерінің жұмысын басқарады.

c) Windows 8.1

d) Windows 10

e) Windows 9

28. Жаңа құжатты немесе кітапты құру үшін қандай пернелерді басу қажет?

a) Ctrl+N

b) Tab

c) Ctrl+m

d) Ctrl+c

e) Ctrl+a

29. Жүйелік программа деген не

a) Компьютердің барлық компоненттерінің жұмысын басқарады

b) Ондаәртүрлі программалар орындалады

c) Кез-келген мәліметтер өңделіп түзетеді

d) бұл терезенің периметр бойынша өтетін тік және көлденең сызықтар

e) бұл терезеде негізгі қолданбалы программалар жұмысы атқарылады

30. Компьютерде рұқсат етілмеген әрекетті орындауға арналған және басқа бағдарламаға ендірілген бағдарламалық код

a) вирустар

b) ворд

c) файлдар

d) музыка

e) диск

31. Кітап, дискета, қатты дискілер қызмет етеді

a) файлдарды жоюға

b) файлдарды сақтауға

c) ақпаратты сақтауға

d) вирустарды сақтауға

e) троян вирусын жоюға

32. Компьютердің жұмыс өнімділігі неге байланысты:

a) вирусқа

b) видеокартаға

c) процессорға

d) процессор жиілігіне

e) файлдарға

33. Көптеген математикалық және логикалық операцияларды есептеуді орындайтын негізгі микросхема

- a) вирусқа
- b) видеокартаға
- c) процессор
- d) процессор жиілігіне
- e) файлдарға

34. Компьютерге ақпаратты еңгізу құрылғысы – бұл

- a) вирусқа
- b) видеокартаға
- c) пернетақта
- d) процессор
- e) троян

35. Жүйелік мәзір қай жерде орналасқан

- a) вирусқа
- b) видеокартаға
- c) тақырып қатарында
- d) процессор
- e) троян

36. Интернет адрестер жүйесінде ҚР көрсететін доменнің аты

- a) kz
- b) ru
- c) az
- d) br
- e) uk

37. Жүйе бұл:

- a) Өзара байланысты элементтердің жиынтығы
- b) Пернетақта
- c) Файлдық вирустар
- d) Процессорлар
- e) Тақырып қатары

Қолданылған әдебиеттер

1. Эндрю Таненбаум “Современные операционные системы”. Санкт-Петербург. Издательство «Питер». 2004.
2. Ф. Новиков, А. Яценко “Операционные системы”. Санкт-Петербург. Издательство «Питер». 2001.
3. Гук М. Аппаратные средства IBM PC. Энциклопедия. Санкт-Петербург. Издательство «Питер», 2000.
4. Accetta, M., Baron, R., Golub, D., Rashit, R., Tebanian, A., and Young, M.: <<Mach: A New Kernel Foundation for UNIX Development>>, Proc. Summer 1986 USENIX Conf., USENIX, pp.112, 1986.
5. Abams, G. B. Iii Agrabal, D. P., and Segiel, H. J.: <.A Survey and Comparison of Fault- Tolerand Multisage Interconnestion Networks>>, Computer, vol.20, pp. 27, June 1987.
6. Adams, A., and Sasse, M. A.: <<Taming the Wolf in Sheep's Clothing>>, Proc. Seventh Int'l Con.on
7. Эндрю Таненбаум “Современные операционные системы” 2-е издание. Москва. Питер-2004.
8. Ф. Новиков, А. Яценко “Операционные системы”. Санкт-Петербург, 2001
9. Волобуева О. П. Операционные системы. Алматы: КазНТУ, 2001, 225 с.
10. Операционные системы: учебник / С.В. Синицын, А.В. Батаев, Н.Ю. Налютин.- М.: Академия, 2010.- 297 с.- (Высш. проф. образование. Информатика и вычислительная техника).
11. Сетевые операционные системы: учебник для вузов / В.Г. Олифер; Н.А.Олифер.- 2-е изд.- М.; СПб; Нижний Новгород: Питер, 2008.- 668 с.- (Учебник для вузов).
12. Операционные системы: разработка и реализация/ Э. Таненбаум; А.Вудхалл.- СПб, : Питер, 2006.- 576с.- (Классика CS).
13. Операционные системы/ А.В Гордеев.- 2-е изд.,.- СПб.; : Питер., 2007.- 416с.
14. Гордеев А.В., Молчанов А.Ю. Системное программное обеспечение. – СПб.: Питер, 2011
15. Кейт Хэвиленд, Дайана Грэй и др. Системное программирование UNIX. – М.: Москва, 2000
16. Глушаков С.В и др. Сетевые технологии WINDOWS NT. – Харьков: М., 2001
17. Стивенс У. UNIX, взаимодействие процессов. - М.: Питер, 2002
18. Гордеев А.В., Молчанов А.Ю. Системное программное обеспечение. -СПб.: Питер, 2001.