

004
O-63

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
С. ТОРАЙҒЫРОВ АТЫНДАҒЫ ПАВЛОДАР МЕМЛЕКЕТТІК УНИВЕРСИТЕТІ

ОПЕРАЦИЯЛЫҚ ЖҮЙЕЛЕР

050704 «Есептеу техника және бағдарламалық қамтама»
мамандығының студенттеріне арналған
оқу-әдістемелік құралы



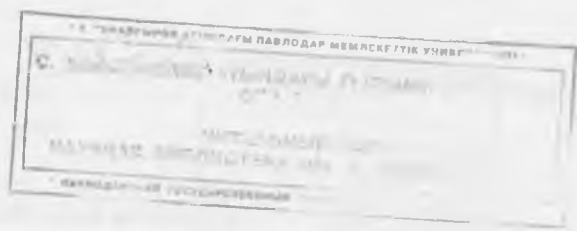
Павлодар

004
0-63

Қазақстан Республикасының Білім және ғылым министрлігі
С. Торайғыров атындағы Павлодар мемлекеттік университеті
Физика, математика және ақпараттық технологиялар
факультеті
Есептеу техникасы және бағдарламалау кафедрасы

ОПЕРАЦИЯЛЫҚ ЖҮЙЕЛЕР

050704 «Есептеу техника және бағдарламалық қамтама»
мамандығының студенттеріне арналған
оқу-әдістемелік құралы



Павлодар
Кереку
2010

ӘОЖ 004.451(075.8)

КБК 32.973.26-018.2я7

И83

С. Торайғыров атындағы Павлодар мемлекеттік университетінің физика, математика және ақпараттық технологиялар факультет кеңесімен басуға ұсынылды

Нікірсарапшылар:

Утегулов Б. Б. – техника ғылымдарының докторы, профессор;

Потапенко О. Г. – техника ғылымдарының кандидаты, профессор.

Құрастырушы Исабеков Ж. Б.

И83 операциялық жүйелер 050704 «Есептеу техника және бағдарламалық қамтама» мамандығының студенттеріне арналған оқу-әдістемелік құралы / құраст. : Ж. Б. Исабеков – Павлодар : Кереку, 2010. – 70 б.

Оқу-әдістемелік құралында операциялық жүйелер барлық формасы баяндалған. Материал мысалдар мен тапсырмалардың үлкен көлемімен түсіндіріліп, сипатталған.

Оқу-әдістемелік құралы «есептеу техника және бағдарламалық қамтама» мамандығының студенттеріне арналған оқу құралы

С.Торайғыров
атындағы ПМУ-дің
академик С.Бейсенбаев
атындағы ғылымдар

КІТАПХАНА

ӘОЖ 004.451(075.8)

КБК 32.973.26-018.2я7

© Исабеков Ж. Б., 2010

© С. Торайғыров атындағы ПМУ, 2010

Материалдық дурыс болуына, грамматикалық және орфографиялық қателерге авторлар мен құрастырушылар жауапты

Кіріспе

Студенттің пәндік оқу-әдістемелік құралы (СП ОӘК) нақты курстың концепциясының анықтайтын құжат ретінде көрсетіледі. Оқу-әдістемелік құралы типтік бағдарлама негізінде құралды, типографиялық әдіспен шығарылып және кредиттік жүйе бойынша оқытылатын студенттерге арналады. Оқу-әдістемелік құралы студентке пәнді оқыту алдында беріледі, өзін-өзі тексеру үшін тестілік тапсырмалардан, емтихан сұрақтарының тізімінен тұрады. Ұсынылған ОӘК бағалығы оқу уақытын тақырыптар мен оқу сабақтарының түрлеріне бөлуден, дәрісханалық және дәрісханадан тыс уақытта студенттердің өзіндік жұмысын ұйымдастырудан, студенттердің танымы мен шығармашылық қызметін белсендіру және оқу мен зерттеу үрдістерінің арақатынасын қамтамасыз етуден тұрады.

1 Операциялық жүйелердің мәні мен функциялары

1.1 Операциялық жүйелердің классификациясы

Машиналық бұйрықтар дәрежесіндегі көптеген компьютерлер сәулеті қолданбалы бағдарламалармен қолдануға тиімді емес. Мысалы, дискпен жұмыс істеу электрондық компоненттің ішкі құрылысының білуін қажет етеді, дискті айналдыру үшін енгізу бұйрығының бақылаушысы, жолды іздеу және форматтау, секторларды оқу және жазу, т.с.с. Орташа бағдарламалаушыға құрылыстың барлық ерекше жұмысын есепке алу қиын (қазіргі терминологияда құрылыс драйверлерін жасаумен айналысу), онда қарапайым жоғарғы дәрежелі абстракция болу керек. Айталық, дисктің ақпараттық кеңістігін файлдар жинағы деп қарастырайық. Файлды оқу және жазу үшін ашуға болады, ақпаратты алу немесе алмау үшін қолдануға, содан кейін жабуға болады. Бұл диск айналдырғышын жылжыту бөлшектерін қадағалаудан немесе мотор жұмысын ұйымдастырудан концептуалды жеңіл. Сол сияқты, қарапайым және түсінікті абстракция көмегімен, бағдарламалаушыдан үзіліс ұйымының барлық керек емес толық мәліметтері жасырылады, таймер жұмысы, жадыны басқару т.с.с. Қазіргі есептеуіш кешендерінде операциялық жадының және процессорлар санының иллюзиялық шексіз өлшемі жасауға болады. Бұның барлығымен операциялық жүйе айналысады. Сол себептен, операциялық жүйе қолданушыға виртуал машина түрінде көрсетіледі, өйткені онымен тікелей компьютер құрылысына қарағанда жұмыс істеу оңай.

Операциялық жүйе компьютер сәулетінің өте қиын бөліктерін басқаруға арналған. Мысалы, егер бір компьютерде бірнеше бағдарлама бір уақытта принтерге шығады деп есептесек, сонда біз жолдар мен парактардың араласқан түрін алушы едік. Операциялық жүйе дисктен баспаға шығару және баспаға шығару кезегін ұйымдастыру арқылы ақпаратты буферлейді. Көп қолданушылы компьютерлерде ресурстарды басқару және оларды қорғау өте қажетті. Демек, операциялық жүйе ресурс менеджері сияқты жадыларды және әр түрлі бағдарламалар арасындағы ресурстарды, процессорларды бақыланатын жіктеу және реттеумен айналысады.

Егер есептегіш жүйе бірнеше қолданушының бірге жұмыс істеуіне жол берсе, онда олардың қауіпсіз жұмыс істеуінің ұйымдастыру мәселесі туады. Ешкім дисктегі ақпаратты жойып немесе басқа да файлдарды бүлдірмес үшін оларды сақтау шараларын қолдану керек. Бір қолданушы бағдарламалары екінші қолданушы

бағдарламаларының жұмысына араласуына рұқсат бермеу керек. Есептегіш жүйені рұқсатсыз қолдануына жол бермеу керек. Осы жұмыстардың бәрін операциялық жүйе қолданушы және оның бағдарламаларының қауіпсіз жұмыс істеуін ұйымдастырушы ретінде орындайды.

ОЖ-ның негізгі функциялары:

- қолданылатын монитормен, оператор бұйрығы ретінде көрсетілген және қажетті тілде жазылған қолданушыдан (немесе жүйе операторынан) бұйрық немесе тапсырма қабылдау;
- бағдарламаларды қосуға, уақытша тоқтатуға, тоқтатуға бұйрықты қабылдау және орындау;
- жедел жадыға орындалуға дайын бағдарламаларды сінтізу;
- бағдарламаны инициациялау (процессор бағдарламаны орындау үшін оған басқаруды беру);
- барлық бағдарламалар мен мәліметтерді идентификациялау;
- басқару жүйесінің файлдармен және мәліметтер базаларымен жұмыс істеуін қамтамасыз ету, осының арқасында барлық бағдарламалармен қамтамасыздандырудың нәтижесі өседі;
- мультибағдарламалау режимін қамтамасыз ету, яғни екі немесе одан да көп бағдарламалардың бір процессорда бір уақытта жұмыс істеу көрінісін тудырады;
- барлық сінтізу/шығару операцияларын басқару және ұйымдастыру функцияларын қамтамасыз ету;
- шын уақыт режимінде жауап уақытына қатал шектелулердің қойылуы (ОЖ-ға сәйкес);
- жадыны жіктеу, көбінесе қазіргі жүйе және виртуал жадының ұйымы;
- берілген стратегия және қамтамасыздандыру тәртібі көмегімен тапсырмаларды диспетчерлеу және жоспарлау;
- орындалып жатқан бағдарламалар арасында мәліметтер алмастыру механизмін ұйымдастыру;
- мәліметтердің сақталуын қамтамасыз ету және бір бағдарламаны екінші бағдарламаның әсерінен қорғау;
- жүйенің істен шығу кезінде оған көмек көрсету;
- қолданушылар өз бағдарламаларын жасауы үшін бағдарламалау жүйесінің жұмысын қамтамасыз ету.

Қорытындылай келіп, ОЖ – бұл бағдарламалар кешені:

- 1) Қолданушы мен компьютер арасындағы интерфейсті;
- 2) ЭЕМ-нің есептеу кезінде қолданылатын ресурстарын және процесстерінің басқарылуын қамтамасыз ету.

Негізгі ресурстар:

- процессор;
- жедел жады;
- периферияді құрылғы.

Ресурстарды басқару мынаған әкеледі:

- 1) ресурстарға жету жолын жеңілдетеді;
- 2) бәсекелес процестердің арасында ресурстарды бөлуді жеңілдетеді.

ОЖ-ның топталуы:

1) қолданушының санына қарай бір қолданушы және көп қолданушы болып екіге бөлінеді;

2) тапсырмаларды дайындау режимі бойынша ОЖ бірбағдарламалық және мультибағдарламалық режим болып бөлінеді;

3) қолданушының ЭЕМ-ге енуінің түрі бойынша:

- a) пакеттік дайындау жүйесі;
- b) уақытты бөлу жүйесі;
- c) шын уақыт жүйесі.

4) есептегіш құрылғысы жүйесінің ресурстарының басқарылу әдістерінің түрі бойынша бір процессорлық және көп процессорлық, желілік және жіктелінген болып бөлінеді.

5) негізгі сәулеттік принцип бойынша микроядерлік және монолиттік болып екіге бөлінеді.

6) ОЖ-ның арналуы бойынша әмбебаптық және арнайы болып бөлінеді. Арнайы ОЖ (ОЖ-ның арнайы тағайындалуы) тіркелген бағдарлама жиынтығымен жұмыс істейді және келесі түрге бөлінеді: тасымалданатын микрокомпьютерлерге және әртүрлі енгізілім орнатылған жүйелерге, ұйымдастырулар және мәліметтерсіз енгізуге, нақты уақыт есебін ішуге және т. б.

7) әр уақытта есептеуіш жүйенің жадында болатын, енгізілетін ОЖ және жүйені енгізу әдісі бойынша көрсетуге болады. Оларды арнайы құрылғы жұмысын басқару үшін қолданады.

Бір процессорлы есептеуіш жүйеде бір уақытта бірнеше бағдарламалар орындалғанда мультибағдарламалау – есептеу ұйымдастыруының әдісі болып түсіндіріледі. Бағдарлама ішіміңдегі әрбір тоқтау (мысалы, мәліметтерді енгізу/шығару операциясының орындалуы) басқа бағдарламалардың орындалуына қолданылады. Мультиесептеу и мультибағдарламалау режимдерін айыру қажет. Мультибағдарламалау режимі бірнеше қосымшаның параллель орындалуын қамтамасыз етеді және осы бағдарламаларды жасаған программисттер, олардың параллель жұмыс істеуінің ұйымдастыру механизміне көңіл аударуы қажет емес. Бұл функцияларды ОЖ

орындайды; ол есептеуіш ресурстарды орындалатын қосымшалар арасында жіктейді, қажетті есептеу және қарым-қатынас синхронизациясын жүзеге асырады. Мультисесептеу режимінде, керісінше, қолданбалы программистер параллель орындалуға және қосымшалар қарым – қатынасына көңіл бөлуі қажет. Қазіргі ОЖ дербес компьютер (ДК) үшін мультибағдарламалау және мультисесептеу режимдерін ұсынады.

Бір есептеуіш жүйелі мультитерминалды ОЖ-да бір уақытта бірнеше қолданушылар, әрқайсысы өз терминалынан жұмыс істей алады. Әр қолданушы өзінде жеке есептеуіш жүйе бар деп ойлайды, яғни иллюзия пайда болады. Есептеуіш жүйеге мультитерминалды қол жеткізу ұйымдастыру үшін мультибағдарламалау жұмысын қамтамасыз ету керек. ДК үшін ОЖ-ның мысалы ретінде Linux-ті алуға болады.

Пакетті оңдеу операциялық жүйе бір есептен екіншіге ең аз ауысуды (минимум) қамтамасыз етеді. Орталық процессор (ОП) уақытын жоспарлау алгоритмі бұл жағдайда төмендегідей: егер ОП – дің біреуіне бөлінсе, онда ол процесс ОП келесі жағдай туғанда қамтып тұрады:

- 1) сыртқы құрылғыға назар аудару.
- 2) процесстің аяқталуы.
- 3) процесстің цикленуінің тіркелген факті.

Осы жағдайдың біреуі пайда болған кезде, басқару басқа процесске беріледі.

Уақытты жіктеу операциялық жүйесі қолданушының сұрағына жүйе реакциясының уақыты минимумға дейін азайтылатын режимде жұмыс істейді. Аз уақытта жауап алғаннан кейін қолданушыда иллюзия пайда болады, яғни ол бүкіл жүйе оған берілген деп ойлайды. Жұмыс келесі түрде орындалады. Осы уақытта ОП алып тұрған процесске, келесі жағдай пайда болғанша оны қамтып тұрады:

- 1) айырбастауға назар аудару;
- 2) процесстің аяқталуы;
- 3) берілген процесске бөлінген уақыт қвантын жоққа шығару.

Нақты уақыт операциялық жүйелері (НУОЖ) түскен тапсырманы берілген уақыт интервалы ішінде өндеуді қамтамасыз етеді, ол увкты интервалын арттыруға болмайды. Тапсырма ағымы жоспарланбаған және оператормен қалыпқа келтірілмейді, яғни тапсырма кез келген болжамсыз уақытта және кезексіз түседі. ОЖ-да, нақты уақытта шешімі жоқ есепке, артық процессорлық уақыт шығыны беріледі (бұл орындалғанда, ОЖ қолданушылардың тілегін өз есебін шығаруда жүзеге асырады, жедел жадыға керек

бағдарламаларды енгізеді және олардың орындалуына қажетті ресурстарды бөледі). НУОЖ-да мұндай шығындар болмауы мүмкін, себебі есептер жиынтығы анықталған және есеп туралы ақпарат сұрақ түскенге дейін белгілі болып тұрады. Нақты уақыт режимін нақты жұмсау үшін мультибағдарламалауды ұйымдастыру қажет (бірақ бұл жеткілікті емес). Мультибағдарламалау есептеуіш жүйенің өндіріс көрсеткішін көтеретін негізгі құралы болып келеді, ал нақты уақыт есеңтерін шығару үшін өндіріс көрсеткіші қажетті факторы болып табылады. Өндірістік көрсеткіштің ең жақсы сипаттамалары нақты уақыт жүйелері үшін біргемерипалды НУОЖ-мен қамтамасыз етеді. Мультитерминалды режимін ұйымдастыру әдістері әрқашан жүйенің жұмысын толығымен баяулатады, бірақ жүйенің функциялы дәрежелерін кеңейтеді. ДК үшін көп танымалы НУОЖ-ның бірі ОЖ QNX болып келеді.

Негізгі сәулеттік принципі бойынша ОЖ микроядерлік және монолиттік болып екіге бөлінеді. Бұл бөліну шартты түрде, бірақ мысал ретінде микроядерлік ОЖ – осы заманғы операциялық жүйелер (ОЖОЗ) QNX және монолиттік ОЖ - Windows 95/98 немесе ОС Linux көрсетуге болады.

Негізгі әдебиет: [1, б. 11-16], [2, б. 19-38], [3, б. 31-137].
Қосымша әдебиет: [9, б. 21-24, 48-51].

Бакылау сұрақтары

1. Операциялық жүйенің анықтамасы?
2. Операциялық жүйенің негізгі тапсырмалары?
3. Қандай параметрлерге байланысты ОЖ топталынады?
4. Мультиесептеу мен мультибағдарламалау жүйелері арасындағы айырмашылық?
5. ОЖ жұмысының пакетті режимі деген не?
6. Уақытты жіктеу ОЖ қандай режимде жұмыс істейді?
7. Нақты уақыт операциялық жүйелеріне қандай жүйелер кіреді?

2 Файлды жүйелер

2.1 Негізгі анықтамалар. Ұйымдастыру принциптері

Файл – мәліметтер жинағы. Файл түсінігімен төмендегі қасиеттер тығыз байланысты:

1) файл – атауы бар бір объект, ол құрамындағы мәліметтермен операциялар жасауға мүмкіндік береді. Негізінде файл атауы бірнеше

символдар жинағы, оның ұзындығы операциялық жүйеге байланысты.

2) файлдың орналасқан жерінен тәуелсіздігі. Бір файлмен жұмыс істеу үшін оның сыртқы құрылымда орналасқан жерін білу қажет емес.

3) енгізу/шығару функциялар жинағы. Негізінде әр операциялық жүйе файл алмастыруды қамтамасыз ететін функциялар жинағымен анықталады. Осы функциялар жинағы мыналардан тұрады:

а) жұмыс істеу үшін файлды ашу. Негізінде бұрыннан бар файлды немесе жаңа файлды ашуға болады. Онда «файлды не үшін ашамыз?» деген сұрақ туу мүмкін. Неге файлды бір уақытта оқып және жазуға болмайды? Негізінде операциялық жүйеге файл тек бір процесспен жұмыс істейтіні туралы орталықтандырылған бұйрық әдісі бар. Ал, ол осы мәліметтерден бір шешім қабылдайды (мысалы, басқа процесстердің файлға снуіне жол бермеу).

б) оқу/жазба. Негізінде файлдармен алмасу кейбір мәліметтер блогымен жүзеге асады. Оқу/жазба функциясында алмастыру мәліметтер блогының өлшемі және оқуға, жазуға арналған мәліметтер блогының саны көрсетіледі. Мәліметтер блогының өлшеміне байланысты алмастыру эффекті жоғарлайды.

в) файл көрсеткішін басқару. Әр алынған файлмен файл көрсеткіші түсінігі байланысады. Бұл көрсеткіш уақыт әр моментінде алмастыруға болатын файлдардың орнын көрсетіп тұрады. Көрсетілген блогпен файл алмастырылған соң көрсеткіш келесі блокқа көшеді. Файлмен жұмыс істеу үшін осы көрсеткішті басқарып білу керек. Мұнда файл көрсеткішінің басқару функциясы бар, ол көрсеткішті файл бойынша көшіруіне рұқсат береді. Көрсеткіш құрамында бір айнымалы бар, ол файлдың ашу функциясымен байланысты.

г) файлды жабу. Бұл операция екі операциядан тұрады: файл құрамын сақтау және жабу немесе файлды жою.

Файлды жүйе дисктегі, не басқа да бір мәлімет жинаушыда мәліметтерді ұйымдастыру әдісін анықтайды. Мысал ретінде қазіргі компьютерде қолданылатын FAT файл жүйесі.

Қазіргі ОЖ-да файлдарды басқару жүйелері бар. Файлдарды басқару жүйесінің арнаулы — мәліметтерге ыңғайлы сну жолдарын ұйымдастыру.

Біріншіден, файл басқару жүйесі арқылы жүйенің мәліметтер бағдарламалары бір-бірімен байланысады. Екіншіден, осы жүйе көмегімен дискідегі кеңістікті орталықты жіктеу және мәліметтерді басқаруға қиындықтар шешімді. Үшіншіден, осы файлдарды басқару жүйесі көмегімен төмендегідей іс-әрекеттерді орындауға болады:

– аты бар мәліметтер жинағын (аты бар файлдар) өз бағдарламаларынан немесе арнайы бағдарламалар басқару әдісімен жасау, жою, атын өзгерту (және басқа операциялар), олар қолданушының өз мәліметтерімен және файлдарды басқару жүйесін активті пайдаланушылардың интерфейс функцияларын нақтылайды;

– дискілі емес перифериялы құрылғылармен файлдар сияқты жұмыс істеу;

– файлдар арасында, құрылғылар арасында және файл мен құрылғы арасында (және керсінше) мәліметтер алмастыру;

– файлдарды басқару жүйесі модулі бағдарламасының көмегімен файлдармен жұмыс істеу (API жартысы файлдармен жұмыс істеуге арналған);

– файлды санкционерлі емес қол жеткізуден қорғау.

Осылайша, «файл жүйесі» - бұл файлдарға жиналған мәліметтерді алу принциптері. Ал, «файлдарды басқару жүйесі» термині - бұл операциялық жүйеде файлдармен жұмыс істеуді қамтамасыз ететін бағдарламалық модульдер кешені (FAT принципі бар және Windows 95/98, Windows NT және т. б. арналған файлдарды басқару жүйелері бар).

Файлдармен жұмыс істегенде ереже бойынша құрылым механизмі енгізіледі және иерархиялық қатынастар ұйымдастырылады. Сол үшін каталог (directory) түсінігі енгізіледі. Каталог файл түрінде ұйымдастырылған мәліметтер туралы ақпарат жинайды. Басқаша айтқанда, каталог құрамында файл дескрипторлары болу керек. Егер файлдар блокты құрылымда жиналса, онда осы каталог көмегімен файлдарды басқару жүйесі керек мәліметтер орналасқан блоктың адресін табады. Каталог ретінде тек қана арнайы, жүйелі ақпараттың құрылымы қолданылмайды (оны кей кезде түбірлік каталог деп атайды), жай файлда каталог бола алады. Мұндай файл-каталогтың арнайы жүйесі болуы керек; файлдарды басқару жүйесі оны басқа файлдар арасында бөліп көрсетіп тұруы керек. Файл-каталогты көбінесе подкаталогом (subdirectory) деп атайды. Егер файл-каталог құрамында басқа да файлдар туралы ақпараттар болса, онда оның ішінде файл-каталогтар болуы мүмкін, нәтижесінде біз шексіз иерархия құруына мүмкіндік аламыз (мұндай файлды объектіні енгізу (файл-каталог), файлдар жүйесін құруына және түбірлік каталогтағы элементтердің шектелген саны қиындықтарын шешуге мүмкіндік береді. Файл-каталогта элементтер санын шектеуге болмайды, сондықтан үлкен өлшемді каталог жасауға болады).

Қол жеткізу принципі (әдісі) дискте сақталған мәліметтерге

нақты қол жеткізуді ұйымдастырады. Файлдар ақпараттарының жинағында жиналуы екіге бөлінеді:

– файлдардың байланысқан жіктеуі (связные распределения файлов);

– файлдардың байланыссыз жіктеуі (несвязные распределения).

Байланыспен жіктеу кезінде файлдық ақпараттың элементтері физикалық жинағында орналасады. Мұндай әдіс үзіліссіз файлдарға ғана, ол қол жеткізудің аз уақытымен және диск кеңістігін дұрыс пайдалануымен сипатталады.

Негізгі кемшіліктері – файлды кеңейту қиындығы және бөлек элементтеріне қол жеткізу қиындығы.

Жазба жүргізу және диск кеңістігін пайдалану секторлар дәрежесінде жүргізіледі. Ең пайдалысы файлдарды блокты жіктеу стратегиясы болып табылады. Бұл жағдайда секторлар тіркелген және айнымалы өлшемді блоктарға жиналады.

Байланыссыз жіктеуде файлға элементтері бар жады беріледі. Мұндай беріліс секторлар мен блоктар болуы мүмкін. Мұндай жағдайда басқару жүйесі бос блоктар мен секторлар тізімін жүргізеді. Әрбір беріліс кезінде блоктан бос шығарылу, және көрсетілген сектордың (блоқтың) сұратылған файлға қосуы орындалады.

Бос тізім азайған сайын жады толған кезде блокировка фазасына жетеді. Жадының босатылуы пайдаланылған сектордың бос орын тізіміне қайтып келуіне әкеледі. Бұл жағдайда әр блок өзінің алғанқы адресімен сипатталады, яғни бірінші сектор адресімен және өлшемімен.

Файлдық блокты ұйымдастыру әдістері:

1) блокты-кезекті құрылым (блокты тізбек);

Ерекшеліктері:

а) мәліметтер жинағымен байланысты адресі бөлігі;

б) осыған байланысты, келесі блокты табу үшін, оны жедел жадыға енгізу керек;

в) манипуляция қарапайымдылығы.

Негізгі кемшілік болып кезектелген блок ішінде блокқа тікелей жолдың болмауы.

Осы кемшілікті жою үшін блок-индексті жіктеу әдісі қолданылады.

2) файлдық блок-индексті жіктеу (индексті блок тізбегі);

Әр блокқа адресіне индексі кестенің номерін және кестедегі индексі көрсету арқылы жүргізіледі.

Артықшылығы:

а) Адрестерді және мәліметтер жинағын тарату, нәтижесінде мәліметтер блогсыз адрестерді өзгертіп отыруға болады (дисктік пен жедел жады арасында араластыру);

б) Көрсетілген индексті кестеде файлдың кезектеп орналасу мүмкіндігінің болуы;

в) Тікелей адресациялау немесе бөлек мәліметтер блогына шығу (файл жазбаларына).

Блоктың берілу ережесі төмендегідей:

1) файлға бірінші блок беріледі және осы блоктың номері директорияға (каталогқа) жазылады. Файлдың көлемі үлкейген сайын, файлға қосымша бос блоктар беріледі және бір блок әртүрлі файлдарда қолданылмайды;

2) FAT жүйесінде диск кеңістігін жасау және басқару әдістері;

3) дисктің сыйымдылығы ОЖ қасиеттеріне және дисковод параметрлеріне байланысты. Структура тұрақты және бір. Мәліметтер концентрлік жолдар түрінде диск бетіне жазылады. Дисктің бір жағындағы ақпараттың мөлшері жолдың санына бір жолдағы секторлаудың жалпы өлшеміне тәуелді. Қатты дисктегі цилиндр – бүлтік орналасқан барлық жолдар.

Форматтау үрдісі екі бөлек операциядан тұрады - жоғарғы және төменгі дәрежеде форматтау: физикалық және логикалық. Физикалық төменгі дәрежеде форматтауда дискке жолдың және сектордың физикалық орындарын көрсету үшін электрондық таңба қойылады. Қатты дискпен жұмыс істегенде бағдарламалаушылар логикалық форматтауды қолданады, өйткені физикалық форматтау көрсетіліп тұрмайды және дисктің микропроцессорымен қол жеткіземіз. Физикалық форматтауды төменгі дәрежеде жұмсақ дискімен жұмыс істегенде қолданамыз. Сектордың дискідегі физикалық адресі сызықтық адрес форматымен (LBA) немесе цилиндр, головка, сектор CHS форматымен берілуі мүмкін. Қатты дисктер үшін шекті мөндер:

1) $S_{max}=65536$ (шеттен центрға дейінгі сектор номерлері);

2) $H_{max}=16$ (BIOS стандартты функцияларын қолданғанда: $S_{max}=1024$ $S_{max}=63$);

3) $S_{max}=255$;

4) ATA = типті дисктің максималды көлемі = дисктегі секторлар санының LBA режиміндегі 128 Гбайт және CHS форматтағы 127.5 Гбайт сектор өлшемінің көбейтіндісіне тең.

SCSI интерфейсі дисководтың мәліметтерді сызықтық адресациялауға мүмкіндігі бар және 2^{32} секторға дейін адресациялайды, яғни 2 Тбайт мөлшерге дейін ақпаратты сақтай алады. Дисктің логикалық структурасы екі қадам бойынша жасалады:

1) бірінші, диск бөлшектерге бөлінеді және әр ОЖ мәліметтер мен бағдарламаларына бір бөлшекпен беріледі;

2) одан кейін әр бөлшек өз ОЖ әдістерімен белгіленеді.

ОЖ жинағышта ақпаратты сақтаудың ұйымдастыру әдістерін таңдайды, ол қолданушының шешіміне байланысты.

FAT 12., FAT 32, NTFS файл жүйелері ең көп таралған. АТ компьютерлері үшін жұмсақ дисктер FAT 12 құрылымы бар бір логикалық диск түрінде ұйымдастырылған.

2.2 Файлдар жүйесі. FAT типті файлдар жүйесі бар логикалық диск (том)

FAT типті файлдар жүйесі бар логикалық диск (том) төрт негізгі облыстардан тұрады:

- 1) резервтік облыс;
- 2) FAT 1 және FAT 2 файлдары орналасқан кестелер облысы;
- 3) түбірлік каталог облысы (FAT 32 – де жоқ);
- 4) файлдар мен каталогтар облысы.

Логикалық диск жүйесінде мәліметтерді ұйымдастыру

FAT12 FAT16	FAT32
Енгізу секторы	Енгізу секторы
FAT1	FS INFO құрылымы
FAT 2	
Түбірлік	Енгізу секторының көшірмесі
Мәліметтер облысы	
	FAT1
	FAT 2
	Мәліметтер облысы

Логикалық дисктің бірінші секторында (сектор №1) – енгізу бағдарламасы (Boot strap, Boot sector (BS)) және (BIOS Parameter – Block (BPB)) параметрлер блогы BIOS. Енгізу секторының алғашқы участогы FAT-тың барлық түріне бірдей (кесте 1).

2.1 Кесте – Енгізу секторының алғашқы аймағы

Элементтің аты	Ығысу	Өлшемі	Сипаттамасы
BS_jmp Boot	00h	3	Енгізу кодына оту инструкциясы (jmp)
BS_OEM Name	03h	8	Текстік жол-фирма, ОЖ версиясы №
BPB_byts Per Sec	0Bh	2	Сектордағы байт саны
BPB_Sec Per Clus	0Dh	1	Кластердегі сектор саны
BPB_RSVD Sec Cnt	0Eh	2	Резервтік облыстағы резервтік сектор саны
BPB_Num Fats	10h	1	FAT көшірмесінің кесте саны (бөлшекте әрқашан екеу)
BPB_Root Ent Cnt	11h	2	FAT 12, FAT 16: үшін түбірлік каталогтағы файлдардың 32 байтты дескрипторлар саны. FAT 16 = 512 жазбасы орындалғанда, FAT 32 = 0 үшін
BPB_Tot sec 16	13h	2	Бөлшектегі секторлардың жалпы саны. Егер берілген өріс 0-ге тең болса, онда сектор саны BPB_Tot Sec 32 өрісінде беріледі
BPB_Media	15h	1	Ақпарат жинағын түрі
BPB_FAT Sz 16	16h	2	FAT 12, FAT 16 үшін – бұл бір FAT көшірмесінде орналасқан сектор саны
BPB_Sec Per Trk	18h	2	Жолдағы сектор саны
BPB_Num Heads	1Ah	2	Бастар саны
BPB_Midd Sec	1Ch	4	Көрсетілмеген сектор саны, бөлшеуге жағдайғын, жинағын үшін поль.
BPB_Tot Sec	20h	4	Бөлшектегі секторлардың жалпы саны. Егер сектор саны 65535-тен асса, осы элемент қолданылады.

2.2 Кесте – Ығысу 24h басталатын FAT 12 және FAT 16 үшін енгізу/шығару секторының құрылымы

Элемент атауы	Ығысу	Өлшемі	Сипаттамасы
BS_Drv Num	24h	1	Дисковод номері жұмсақ дискі үшін 0-ден 3-ке дейін, ал қатты диск үшін 20h басталады
BS_Reserved	25h	1	Windows NT үшін бекітілген
BS_Boot Sig	26h	1	Енгізу жобасының келесі көрсеткіші. Келесі 3 өрістің бар екендігін көрсетеді (29h)
BS_Vol ID	27h	4	Логикалық диск номері
BS_Vol Lab	2Bh	11	Диск таңбасы
BS_Filsys Type	36h	8	Файлды жүйе түрі

2.3 Кесте – Ығысу 24h басталатын FAT 32 үшін сектордың құрылымы

Элемент атауы	Ығысу	Өлшемі	Сипаттамасы
BPB_FAT_Sz 32	24h	4	FAT бі көшірмесінде орналасқан секторлар саны
BPB_Ext Flags	28h	2	Активті FAT номері
BPB_Fs_Ver	2Ah	2	FAT 32 версиясының номері
BPB_Boot Clus	2Ch	4	Түбірлік каталогтың бірінші кластері үшін кластер номері (2)
BPB_FSINF O	30h	2	Логикалық дисктің резервті облысындағы FS INFO құрылымындағы сектор номері (1)
BPB_BK Boot_Sec	32h	2	Енгізу секторының резервті көшірмесі бар сектор номері (6)
BPB_Reserve d	34h	12	Резервтелінген
BS_Drv_Num	40h	1	Дисковод номері
BS_Reserved 1	41h	1	
BS_Boot_Sig	42h	1	
BS_Vol_ID	43h	4	Логикалық диск номері
BS_Vol_Lab	47h	11	Таңба
BS_File_Sys Type	52h	8	Файлды жүйе түрі
	1FE		Код 55h
	1FF		Код AAh сигнатура

FSINFO құрылымы резервті облыстың 01 секторында орналасады және дисктегі бос кластер соңы туралы және FAT кестесіндегі бос кластердің номері туралы ақпаратқа ие болады.

2.4 Кесте – FSINFO құрылымы

Элемент атауы	Ығысу	Өлшемі	Сипаттамасы
FSI_Lead_Sig	000h	4	416152 52h құрылымы - FS INFO белгісі
FSI_Reserved1	004h	480	Резервтелінген
FSI_Struct_Sig	1F4h	4	614172 72h – сигнатура
FSI_Free_Count	1E8h	4	Бос кластер ағым саны. Егер ол сан 0FFFFFFF болса, онда бос кластер саны белгісіз, оны есептеп табу керек
FSI_Nxt_Free	1ECh	4	Кластер номері бар, осы арқылы дискті драйвер бос кластерді табады. Егер 0FFFFFFF болса, онда бос кластер номер 2 кластерден басталады
FSI_Reserved2	1F0h	12	Өріс резервтелінген
FSI_Tail_Sig	1FCh	4	Сигнатура – FS INFO соңын көрсетеді

Файлдардың орналасу кестелерінің ішкі ұйымдастыруы және

арналуы.

FAT-ты қолданғанда логикалық дискінің мәліметтер облысы бірнеше бір өлшемді аймақтарға бөлінеді. Ол кластерлер деп аталады.

Кластерлердегі секторлар саны 2^N (1,2, ..., 64) болуы керек. Кластер өлшемі қолданып жатқан FAT жүйесіне және логикалық диск өлшеміне тәуелді.

Кластерлер 2-ден басталып номерленеді. Әр кластерге FAT кестесі бойынша элемент қойылған.

FAT-тың бірінші 2 элементі резервті болып келеді. FAT түрі count of clusters (coc) түсінігімен анықталады:

- 1) егер coc < 4085 --> FAT 12;
- 2) егер 4084 < coc < 65525 --> FAT 16;
- 3) егер 65525 < coc --> FAT 32.

FAT 32-де 4 үлкен екілік разряд резервте болып, ОЖ жұмысына қатыспайды. 1-ші кластердің номері файлды алып тұрған каталогта көрсетіледі.

Каталог – бұл кесте, мұнда дисктердегі файлға 32 байт өлшемді жазба болады.

Түбірлі каталог (Root directory) – дисктің негізгі каталогы, бұдан подкаталогтар тізбегі басталады. FAT 12, FAT 16-да – 512 элементті сақтайтын өлшемді (16 Кб) арнайы орын.

FAT 32-де – шексіз өлшемді файл.

2.5 Кесте – Каталог элементінің құрылымы

Өріс атауы	Ығысу	Өлшемі	Сипаттамасы
DIR Name	00h	11	Файлдың қысқа аты
Dir Attr	0Bh	1	Файл атрибуты
DIR NT Res*	0C'h	1	NT үшін резервтелінген
DIR Crt Time Tenth*	0Dh	1	Жасау уақытының анықталған өрісі
DIR Crt Time	0Eh	2	Жасау уақыты
DIR Crt Date*	10h	2	Жасау датасы
DIR Lst Acc Date*	12h	2	Соңғы қараған дата
DIR FST Clas III*	14h	2	Файлдың помер бірінші кластерінің үлкен сөзі
DIR Wrt	16h	2	Файлда соңғы жазбаның орындалу уақыты
DIR WrtDate	18h	2	Файлда соңғы жазбаның орындалу датасы
DIR Fst Clus Lo	1Ah	2	Файлдың номер бірінші кластерінің кіші сөзі
DIR File Size	1Ch	4	Файлдың байтпен өлшемі

* - тек қана FAT 32 үшін

DIR Name[0] қысқа атауының 1-ші байтты каталогтағы

элементтің бос еместігін көрсетеді.

Егер $DIR_Name[0] = E5h$ болса, онда каталог элементі бос.

Егер $DIR_Name[0] = 00$ болса, онда каталогтың бос облысының басталуы.

Файлдың ұзын аты (LFN). Каталогқа бірінші жазылады. Одан кейін файлдың стандартты сипаттамасы, оның құрамында арнайы алгоритммен қысқартылған осы атаудың варианты. Файлдың ұзын атының белгісі – бұл атрибуттың байт разрядында (0 – 3) бірліктің болуы.

2.6 Кесте – Ұзын атаулы файлдың сипаттамасын каталогқа жазу тәртібі

Оріс атауы	Ығысу	Өлшемі	Сипаттамасы
DIR_Counter	00h	1	Фрагмент номері
Dir_Lname 1	01h	10	Ұзын атау фрагментінің бірінші участоғы
DIR_Attr	0Bh	1	Файл атрибуты
Dir_Flagf	0Ch	1	Жалау байты
DIR_Chk_Sum	0Dh	1	Қысқа атаудың бақылау қосындысы
DIR_Lname 2	0Eh	12	Ұзын атау фрагментінің 2-ші участоғы
DIR_First	1Ah	2	Бірінші кластер номері
DIR_Lname 3	1Ch	4	Ұзын атау фрагментінің 3-ші участоғы

Түбірлі каталогтардан басқа барлық каталогтар мынадай символдарға ". ." немесе ". ." ие болады. Файл өлшемі 32 разрядты санмен беріледі, максималды (max) өлшемі 4Гб. Каталогтың құрамында 65536-ға дейін 32 байтты элементтер болады. Шектеу каталог элементінің сәтінгінің өлшеміне байланысты > 16 разрядтар.

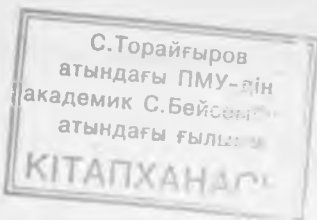
Файлды DEL командасы арқылы жою мынаған әкеледі:

- 1) каталогта бірінші атаулы байт E5-те орналасады;
- 2) осы файлға қатысы бар FAT-та барлық кластерлер босайды.

Кластердегі жазылған мәліметтер өзгермейді (олар тек кластер басқа файлға жазылуға берілгенде өзгереді).

2.3 NTFS тінгі файл жүйесінің логикалық дискі (том)

NTFS файлы жүйесінің толық аты – New Technology File System, яғни жаңа технологиямен жасалған файлы жүйе. NTFS өте сенімді, істен шыққанда орнына қайтадан келе алады. Файл және каталог атын жазу үшін ұлттық алфавитті қолдануға, файлы жинақтауға және шифрлеуге, мәліметтерге шексіз қол жеткізу және т.б. мүмкіндіктер береді. NTFS жүйесі енгізу секторынан басталады.



015249

2.7 Кесте – Енгізу секторының алғашқы облысының форматы

Біттеу, байт	Өлшемі, байт	Сипаттамасы
0	3	JMP xxxx командасы (бастапқы енгізу)
3	8	Операциялық жүйенің фирма-дайындаушысының аты және версиясы
0xВ	2	Байтпен көрсетілген сектор өлшемі
0xD	1	Бір кластердегі сектор саны
0xF	7	Резервтелінген
0x15	1	Мәліметтер жинағыш түрі
0x16	2	Резервтелінген
0x18	2	Жолдағы секторлар саны
0x1A	2	Магнитті головкалар саны
0x1C	8	Резервтелінген
0x24	2	0x80 санына ие болады
0x26	2	0x80 санына ие болады
0x28	4	Бөлімдегі секторлердің жалпы санының кіші сөзі
0x2C	4	Бөлімдегі секторлердің жалпы санының кіші сөзі
0x30	4	Кластер номерінің кіші сөзі, онда MFT кестесі
0x34	4	Кластер номерінің үлкен сөзі, онда MFT кестесі
0x38	4	Кластер номерінің кіші сөзі, онда MFT кестесінің көшірмесі басталады
0x3C	4	Кластер номерінің үлкен сөзі, онда MFT кестесінің көшірмесі басталады
0x40	4	кластердегі MFT жазбасының өлшемі
0x44	4	Кластердегі индекс буферінің өлшемі
0x48	4	Дисктің сериялық номерінің кіші сөзі
0x4C	4	Дисктің сериялық номерінің үлкен сөзі

Енгізу секторының көшірмесі бөлімнің ортасында немесе соңында жасалынады.

NTFS бөлімінің геометриясын анықтау. Бұл бөлімшенің геометриясына келесі параметрлер кіреді:

1) сектордың байттағы мөлшері, кластердің сектордағы мөлшері, MFT басқы файлдар таблицасының бастапқы кластерінің номері және оның көшірмесі. Әдетте, сектордың көлемі қатты дисктер үшін 512 байтқа тең. Басқа деректерді сақтаушылар (компакт-дисктер және магнитооптикалық дискілер) сектордың басқа көлемін қолдана алады. Мысалға, 1024 немесе 2048 байт;

2) кластердің көлемі – NTFS геометриясының маңызды бір

құрамы. Бұл көлем бөлiмдi форматтаған кезде орнатылады және оның ауқымына тәуелдi болады. Егер NTFS жүктеу секторының iшiндегiсi жоғалтылған болса, онда келесi: кластер көлемiнiң NTFS бөлiмiнiң көлемiне тәуелдiлiк кестесiн қолдануға болады;

NTFS бөлiмiнiң көлемi, Гбайт	Кластер көлемi, секторлар 512 байттан
0.5-ке дейiн	1
0.5-тен 1-ге дейiн	2
1-ден 2-ге дейiн	4
2-ден жоғары	8

3) NTFS-тiң физикалық құрамы. Бөлiмiнiң құрамы, басқа да жүйелер секiлдi NTFS барлық пайдaлы орындарды кластерлерге бөледi, кластер – бiр уақытқа қолданылатын деректер блогы. NTFS көбiнесе кластердiң әрбiр көлемiн қолдап тұрады – 512 байттан 64 Кбайтқа дейiн. NTFS дискi шартты түрде 2-ге бөлiнедi. Дискiнiң бастапқы 12%-ы MFT зонасына берiледi, мұнда MFT метафайлы осетiн кеңiстiк орналасқан. Бұл облысқа жазбалар енгiзу мүмкiн емес. MFT зонасы әрдайым бос болып тұрады, бұл – ең басқы, қызметтiк файл (MFT) осiн жатқан кезiнде фрагментталмауы үшiн керек. Қалған 88% - файлдарды сақтау үшiн арналған қарапайым кеңiстiк.

Бiрақта дискiнiң бос орны барлық физикалық бос орындарды қамтиды, оған MFT зонасының толтырылмаған бөлшектерi де кiредi. MFT зонасының қолданылуы келесiдей: файлды қарапайым кеңiстiкке жазуға болмаған жағдайда, MFT зонасы жай қысқарады (ағындағы операциялық жүйелер версияларында дәл 2 есеге), яғни осы арқылы ол файлды жазу үшiн орын босатады. Қарапайым облыста орын босаған кезде MFT зонасы қайтадан кеңейiп мүмкiн. Бұл жағдайда зонада қарапайым файлдар да қалуы мүмкiн.

MFT және оның құрамы. NTFS файлдық жүйесi – структуризациясының ең даңқты нәтижесi: жүйенiң әрбiр элементi файл түрiнде болады, қызметтiк ақпараттың өзi де. NTFS-тегi ең негiзгi файл MFT деп аталады немесе Master File Table – файлдардың жалпы кестесi. Дәл осы файл MFT зонасында орналасқан және дискiдегi қалған файлдардың орталықтандырылған каталог қызметiн атқарады. MFT бекiтiлген көлемдегi (1 Кбайт) жазбаларға бөлiнген және әрбiр жазба белгiлi бiр файлға сәйкес келедi. Бастапқы 16 файлдар қызметтiк түрде болады және операциялық жүйеге тәуелдi емес, олар метафайлдар деп аталады, ең бiрiншi метафайл – MFT-тiң өзi. MFT-тiң осы бiрiншi 16 элементтерi бекiтiлген орны бар дискiнiң жалғыз ғана бөлiгi. Олардың көшiрмесi дискiнiң дәл ортасында

сақталады.

Метафайлдар. NTFS-тің бастапқы 16 файлы (метафайлдар) қызметтік түрде ғана болады. Олардың әрқайсысы жүйе жұмысының әрбір аспектісіне жауап беріп отырады.

Метафайлдар NTFS дискінің түпкі каталогінде орналасқан және «\$» символымен басталады.

2.8 Кесте – Негізгі метафайлдар және олардың қызметі

\$MFT	MFT-тің өзі
\$MFTmirr	Дискінің ортасында орналасқан MFT-тің бастапқы 16 файлдарының көшірмесі
\$LogFile	Журналдауды қолдаушы файл
\$Volume	Қызметтік ақпарат – томның таңбасы, файлдық жүйенің версиясы, т.б.
\$AttrDef	Томдағы файлдардың қалыпты атрибуттарының тізімі
\$.	Түпкі каталог
\$Bitmap	Томдағы бос орынның картасы
\$Boot	Жүктеу секторы (егер бөлім жүктелмелі болса)
\$Quota	Қолданушылардың дисктік кеңістік пайдалану құқығы жазылған файлы (NTFS-тің ішінде ғана жұмыс істей бастайды)
\$Upcase	Ағындағы томдағы файлдардың атындағы бас және кіші әріптердің сәйкестік файл – кесте. NTFS-тегі файлдар аты Unicode жазылған, сондықтан да осы файл-кесте қолданылады.

Файлдар және ағын. Әрбір файл үшін міндетті элемент MFTB-да жазылу. Мұнда файлдар туралы құрамынан бөтен барлық ақпарат сақталады. Файлдың аты, көлемі, дискідегі орны, т.б. Егер ақпарат үшін MFT-тің бір жазбасы жетіспесе, онда бірнешеуі қолданылады. Бірінен соң бірі қолданылуы міндетті емес. Жүздеген файлды қамтып тұрған файлдар негізгі файлдық облыста «физикалық» түрінде болмайды, мұндай файлдың деректері бір орында сақталады – MFT-те.

NTFS-тегі әрбір файлдың тек ағындары (streams) ғана болады. Осы ағындардың бірі мәнді алып жүреді, яғни файлдың деректерін. Файлдың атрибуттары (сипаты) көбінесе тағы ағын болып келеді! Яғни файлдың базалық мәні біреу ғана – MFT-тегі номері, ал қалғандары опционалды. Файлдың аты кез келген символды болуы мүмкін, ішіне ұлттық алфавиттердің толық жиынтығы да кіреді, себебі берілген деректер Unicode-ға көрсетілген. Яғни 16-лық бит түрінде, ал ол өз атынан 65535 түрлі символдарды береді. Файл атының max-ді ұзындығы – 255 символ.

NTFS-тегі каталогтар ерекше файлдар болып келеді. Бұл файлдар

басқа файлдар мен каталогтарға үндеуді сақтайтын және берілген деректердің дискідегі иерархиялық құрылысын құрайтын файлдар. Каталог файлы блоктарға бөлінген және осы блоктардың әрқайсысы каталогтың элементі туралы толық ақпаратты қамтитын файлдың атын, базалық атрибуттарын және MFT элементіне үндеуді (ссылка) ұстап тұрады. Каталогтың ішкі құрылымы бинарлық (екілік) ағаш түрінде болады. Бинарлық ағаш файлдардың аттарын тез арада табу үшін оңай болатындай етіп орналастырады, яғни файлдың орналасуы туралы сұраққа екілік (екі белгішелі) жауап қайтару арқылы файлды табуға көмектеседі. Берілген сұраққа бинарлы ағаш былай жауап қайтарады: берілген элементке сәйкес ізделініп отырған аттың қай топта орналасқан – жоғары не төмен.

Осыған орай, 1000-ған файлдың ішінен біреуін табу үшін, мысалға, FAT файлы, орташа есеппен 500 салыстыру жүргізілуі керек (файлдың ортасында табылу ықтималдылығы көп), ал ағаш негізіндегі жүйе – 12-ге жақын ($2^{10} = 1024$).

Дискі бойынша қарапайым навигация жүргізу үшін, MFT-ке әр файл үшін кіре беру міндетті емес, файлдар каталогындағы файлдар туралы жалпы мағлұматты оқу жеткілікті. Дискінің басқы каталогы – түпкі, оның MFT метафайлының басынан оған арнайы үндеу беріледі.

Негізгі әдебиет: [1, б. 163-208], [1, б. 166-177], [1, б. 247-278], [2, б. 28-55], [2, б. 158-182], [4, б. 56-67]. Қосымша әдебиет: [9, б. 146-191], [9, б. 269-300], [13, б. 224-262], [15, б. 365-383].

Бақылау сұрақтары

1. «Файл жүйесі» түсінігіне анықтама беріңіз.
2. Файлды басқару жүйесінің негізгі міндеттері қандай?
3. Файлмен жұмыс істегенде иерархиялық қатынастар қалай ұйымдастырылады?
4. Жинағышта файлдың байланыспен жіктеуі байланыссыз жіктеуден айырмашалығы қандай?
5. FAT 12, FAT 16 (V FAT) үшін арналған логикалық диск құрылымы FAT 32- ден айырмашалығы неде?
6. Параметрлер блогы BIOS қандай ақпараттарға ие болады (BPB)?
7. FS INFO құрылымының міндеті, оның құрылымы қандай?
8. Ұзын каталогқа файлдың жазылуын көрсетіңіз.
9. NTFS бөлімінің жүктеу секторының бастапқы облысы неден тұрады?

10. MFT бастапқы кластерінің нөмерін және көшірмесін қайдан табуға болады?

11. Қандай жағдайда \$ MFT файлында кеңейтілген жазба жасалады?

3 Қатты дискінің логикалық құрылымы

3.1 Қатты дискідегі деректердің ұйымдастырылуы

Қатты дискідегі кеңістік бір немесе бірнеше бөлімдерге бөлінуі мүмкін. Бүкіл дискілік кеңістіктің бірнеше бөлімдерге бөлінуінің пайдалы себептері бар. Біріншіден, бұл деректердің сақталуын құрамдайды. Мысалға, бөлек бір бөлімді операциялық жүйе бағдарламалық камтамасыздандыру үшін белгілеп болсек, ал басқа бөлімді және қолданушының деректері үшін белгілесек, онда соңғысын жүйелік файлдардан бөлуге мүмкіндік болады. Бұл жүйенің беріктігін арттырып ғана қоймай, оның қызмет етуін қолайлы қылады. Екіншіден, әрбір бөлімде өзінің файлдық жүйесі ұйымдастырылуы мүмкін, көп жағдайларда ол өте керек. Нақты қатты магниттік дискінің (НҚМД) ең бірінші секторында, яғни 0-0-1 координаталы секторда дискінің логикалық ұйымдастырылуы туралы ақпаратты сақтайтын деректер құрылымы орналасқан.

Бұл құрылымның аты – басты жүктеу жазбасы (Master Boot Record, MBR). MBR-де дискінің логикалық ұйымдастырылуын анализдеуге, операциялық жүйенің жүктеу бағдарламасын тауып және оперативті (желел) жадыға жүктеуге болатын бағдарлама бар. MBR-де орналасқан бұл бағдарлама сыртжүйелік жүктеуші (Non-System Bootstrap, NSB) деп аталады. ОХ 1 BE-нің MBR-ге ығысуында бөлімдер кестесі орналасқан (Partition Table, PT), ал 64 байтты үстап, 4 элементтен тұрады. Бұл кестенің әр элементін бір бөлім тізімдейді.

MBR-дің соңғы 2 байты 55AA_(h) түрінде болады, яғни 0 мен 1 кезегесу мәндерінде. Бұл сигнатура ақпаратты тасымалдайтын желілердің жұмыс істеу қабілетін тексеру үшін қабылданған. 55AA_(h) мәні барлық жүктелмелі секторларда бар.

3.1 Кесте - Қатты дискінің 1-ші секторының ішкі құрылымы.

Ығысуы	Көлемі	Бағдалауы
000 - 18Dh	446	Жүктелмелі жазба (NSB)
1BE - 1CDh	16	1-ші бөлім элементі
1CE - 1DDh	16	2-ші бөлім элементі
1DE - 1EDh	16	3-ші бөлім элементі
1EE - 1FDh	16	4-ші бөлім элементі
1FE - 1FFh	2	Сигнатура 0AA55h

Әрбір PT элементінің құрылымы. Бөлімдер кестесінің элементінің 1-ші байты ретінде бөлімнің екпінділік Boot Indicator (0 мәні – екпінді емес, 128 (80_h) – екпінді) жалауы тұрады. Ол берілген бөлім жүйелік-жүктемелі ме, жоқ па екенін анықтайды. Нәтижесінде операциялық жүйенің жүктелу процесі бірінші сектордың өте екпінді бөлімнен жүктелуінен және оның ішіндегі орналасқан бағдарламаға басқаруды беруден басталып, әрі қарай жалғасады. Бөлімнің тек біреуі ғана екпінді болады және ол әдетте MBR-де орналасқан NSB бағдарламасымен тексеріледі.

Бөлімнің екпінділік жалауынан кейін бөлім басталатын баспа нөмері байты тұрады. Одан кейін сектор нөмерін және операциялық жүйені жүктеушінің бірінші секторы орналасқан жүктеуші сектор цилиндрінің нөмерін қамтитын екі байт орналасады. Содан соң System IP (ұзындығы – 1 байт) кодтық идентификатор болады. Ол берілген бөлімнің қай операциялық жүйеге жататынын және осы бөлімде сәйкесті файлдық жүйенің орналастырылуын көрсетеді.

3.2 Кесте – Бөлімдер элемент форматы

Partition Table элементінің жазба аты	Ығлеу	Ұзындық (байт)
Бөлімнің екпінділік белгісі	00h	1
Бөлім басының басты нөмері	01h	1
Бөлімнің жүктемелі секторының цилиндр, сектор нөмері	02h	2
Операциялық жүйенің кодтық идентификаторы	04h	1
Бөлім соңының басты нөмері	05h	1
Бөлімнің соңғы секторының цилиндр, сектор нөмері	06h	2
Бастапқы сектордың нөмеріне байланысты кіші және үлкен сөз	08h	4
Сектор бойынша бөлімнің көлеміне байланысты кіші және үлкен сөз	0Ch	4

3.3 Кесте – Сектор және цилиндр нөмерлері бар жазба форматы

Цилиндр нөмерінің биттары								Сектор нөмерінің биттары							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

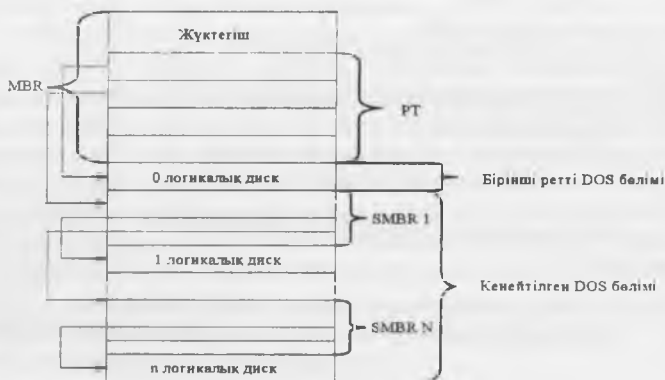
Егер де код өрісінде 0 жазылып тұрса, онда ешнәрсеге бөс болып шығады, яғни ол дискіде ешқандай бөлімді анықтамайды.

Диск бөлімдері 2 типті болуы мүмкін: бастапқы (primary) және кеңейтілген (extended). Бастапқы бөлімдердің тақ-ды саны 4-ке тең. Егер бастапқы бөлімдер бірнешеу болса, онда олардың тек қана біреуі екпінді бола алады. Екпінді бөлімде орналасқан жүктеушіге, компьютер қосылған кезде, сыртжүйелік жүктеуші көмегімен басқару

беріледі. DOS жүйесі және басқа да DOS спецификациясын қолданатын операциялық жүйелер үшін қалған бастапқы бөлімдер бұл жағдайда көрінбейтін болып есептеледі. Мысалға, Windows 9x операциялық жүйесі.

Қабылданған спецификацияларға байланысты бір қатты дискіде тек бір ғана кеңейтілген бөлім болуы мүмкін, ал ол әрі қарай көп бөлімшелерге – логикалық дискілерге (logical disks) бөліне алады. Кеңейтілген бөлімде MBR-дің екінші жазбасы (Secondary MBR, SMBR) бар, оның ішінде бөлімдер кестесінің орнында сәйкес логикалық дискілер кестесі (Logical Disks Table, LDT) орналасқан. LDT бір ғана логикалық дискіден тұратын бөлімнің орналасуын және мінездемесін сипаттайды және де SMBR-дің келесі жазбасын спецификациялайды. Яғни, егер кеңейтілген бөлімде K логикалық дискілер жасалса, онда бұл бөлімде SMBR-дің K нұсқасы болады. Бұл нұсқаулар бір тізімде жиналады. Бұл тізімнің әрбір элементі сәйкес логикалық дискті сипаттап, келесі (соңынан басқасы) элементке үндеп тұрады.

Осылай бастапқы бөлімнің сипаттаушысы логикалық дискінің жүктеуші секторына сілтеп тұрады. Кеңейтілген бөлімнің сипаттаушысы екінші MBR-ге (SMBR) сілтеп тұрады.



Сурет 3.1 – Дискілерді бөлімдерге бөлу

Операциялық жүйенің жүктелуі. Бастапқы жүктеу процедурасы (Bootstrap loader) бағдарламалық үзу ретінде шақырылады (Bios Int 19h). Бұл процедура рұқсат етілген және қатынасуға дайын тұрған бірінші құрылғыны анықтап, одан оперативті жадыға басты бағдарламаны – жүктеушіні жүктеуге тырысады. Қатты магниттік

дискідегі жинақтаушы үшін – ол MBR-ден сыртжүйелік жүктеуші және оған басуарылу беріледі. Басты жүктеуші дискідегі екінді бөлімді анықтап, оның өз жүйелік жүктеушісін жүктеп, басқаруды соған береді. Соңында, осы жүктеуші операциялық жүйенің керекті файлдардан тауын және жүктеп, басқаруды соған береді. Кейін операциялық жүйе өзіне тәуелді бағдарламалық және аппараттық құралдарды инициалдайды. Ол бағдарламалық үзу механизмі арқылы шақырылатын жаңа сервистерді қосып, BIOS-тың кейбір сервистерін кеңейтеді немесе ауыстырады. Қазіргі мультибағдарламалық операциялық жүйелерде BIOS сервистерінің басымы, бастапқысында ПЗУ-да орналасқан, әдетте өзінің ОЖ драйверлерімен ауысады, себебі олар үзу режимде жұмыс істеу керек.

Microsoft операциялық жүйесі әріптерді (диск аттарын) келесідей таниды: А: В: аттары иілгіш дискілерге бекітілген және олардың компьютердің конфигурациясында бар-жоқтығы әсер етпейді. Қатты дискілердің бөлімдері С әріпнен бастап әрі қарай атала береді, басында негізгі бөлімдерге аттар қойылады, ал содан кейін кеңейтілген бөлімдердің логикалық дискілері аталады. Осыған орай, кеңейтілген бөлімдерде жасалған дискілердің нөмері 2-ден басталады.

Негізгі бөлімдерге ат қойылу тәртібі:

- 1) 0 каналының Master disk-ci;
- 2) 0 каналының Slave disk-ci;
- 3) 1 каналының Master disk-ci;
- 4) 1 каналының Slave disk-ci.

Логикалық дискілер атынан басқа (С:, D:, E:, т.б.) бөлімдер нөмеріне ие болады. С: дискі – 1-нөмер, D: дискі – 2-нөмер, E: дискі – 3-нөмер, т.б. (Дәл осы бөлімдер нөмерлері Windows NT/2000/XP жүйелік жүктемесіне операциялық жүйемен таңдалған файлын қай жерде орналасқанын көрсететін boot.ini файлында қолданылады). Дискілерді бөлімдерге бөлетін және FAT, SMBR-ді құрайтын утилиттер F-disk деп аталады.

Бөлімдер кестесін түзететін утилиттер – DISK EDITOR.

Қуатты бағдарламалардың бірі – Partition Magic – қатты дискілердің бөлімдерімен жұмыс істеу үшін арналған.

Қатты дискіні жұмысқа дайындау үшін 3 этапты орындау керек:

- а) төмен дәрежедегі форматтау;
- б) конфигурирлау;
- в) жасалған логикалық дискілерді жоғары дәрежеде форматтау.

3.2 Операциялық жүйедегі енгізу-шығаруды басқару

Операциялық жүйе енгізу-шығару құрылғыларымен тиімді

басқаруды ғана қамтамасыздандыру емес, сонымен қатар енгізу-шығару құрылғыларының қолайлы және тиімді виртуалдық интерфейсіні құру керек. Бір модельде құрылғылардың кеш жиынтығын біріктіре алатын енгізу-шығару жүйесі әмбебап болуы керек. Ол барлық құрылғылардың қажеттіліктерін өтеп тұру керек, қарапайым тышқаннан бастап, клавиатура, принтер, графикалық дисплей, дискілік жинақтауыш, тіпті желіге дейін. Бір жағынан, параллельді орындалып жатқан тапсырмалар үшін енгізу-шығару құрылғыларын қолдануға болатын мүмкіндікті қамтамасыз етіп, құрылғылардың бір-біріне кедергі ету ықтималдығын тоққа шығару керек.

Осыған орай, келесі принцип ең маңызды болып келеді: енгізу-шығарумен басқару бойынша жүргізілетін әрбір операция ерекше жеңілді деп саналады және мұндай операциялар операциялық жүйенің өз кодымен ғана орындалады.

Енгізу-шығару құрылғысы бөлінетін және бөлінбейтін болып бөлінеді. Бөлінетін құрылғыларға магнитті дискідегі жинақтауыш, компакт-дискілерді оқитын құрылғылар жатады. Бөлінбейтін құрылғыларға принтер жатады. Операциялық жүйелер осы құрылғылардың бәрін басқару керек, нәтижесінде қатар орындалып жатқан тапсырмаларға түрлі енгізу-шығару құрылғыларын қолдануға мүмкіндік береді.

Енгізу-шығаруды басқару жүйесінің негізгі қызметтері:

1) нақты сыртқы құрылғылар бойынша бағыттылған функциялардың енгізу-шығаруына сұраныстардың түсініктемесі (интерпретация);

2) нақты құрылғылар бойынша іске асырылған енгізу-шығару операцияларының орындалуын бақылау;

3) енгізу-шығару операциялары орындалып жатқан кезде пайда болған оқиғалар бойынша іске асу;

4) процессор мен сыртқы құрылғылардың арасындағы шартты түрде керекті синхронизация.

Енгізу-шығарудың 2 негізгі режимі бар: енгізу-шығару құрылғысының дайындық туралы сұранысымен айырбастау режимі (бағдарламаланатын енгізу-шығару) және үзуді қолданумен айырбастау режимі.

Бағдарламаланатын енгізу-шығару құрылғының жағдайына байланысты іске қосылған операцияны бағдарламалық бақылауға және тікелей процессорға байланысты ақпаратты жіберу функциясын бақылауға негізделген.

Бұл құрылғы оның айырбасқа дайындығын көрсететін бітпен

сипатталады.

Құрылыстар арасында айырбасты ұйымдастыру үшін процессор мен құрылғының арасындағы ақпараттар айырбасын іске асырып тұратын бір символға немесе бір жолға арналған буферлер қолданылады. Осылай құрылғы буфер толтырылған кезде процессорге хабар бергенде, енгізу құралы өзінің дайын екендігі туралы белгі береді. Ал буфер бос болған кезде өзінің дайын екендігі туралы шығару құралы белгі береді.

Бағдарламалық енгізу-шығару кезінде құрылғының буфері мен процессордың регистрі арасында айырбас жүреді. Оның алгоритмі мынадай (құрылғыға жазу мысалы):

1-қадам: процессор сөзді жедел жадыдан регистрге есептейді;

2-қадам: құрылғыдан оның дайын екендігі туралы белгіні оқу;

3-қадам: белгіні тексеру: егер құрылғы бос емес болса, онда 2-ші қадамға оралу;

4-қадам: егер құрылғы бос немесе дайын болса, онда сөзді процессор регистрінен құрылғының буферіне беру;

5-қадам: «жазу» командасын құрылғыға беру;

6-қадам: бағдарлама параметрлерінің модификациясы: сөздер мекен-жайы санаушының (счетчик) 1-ге өсуі, берілген сөздер санаушының 1-ге келуі;

7-қадам: егер сөздер санаушы 0-ге тең болса, онда 1 қадамға қайта оралу, керісінше болса – 8-қадамға өту;

8-қадам: құрылғының жағдайын есептеу;

9-қадам: егер бос емес болса, 8-ші қадамға оралу, керісінше болса 10-шы қадамға өту;

10-қадам: құрылғыны тазарту.

Процессорды шақыратын бағдарламасымен іске қосылған енгізу-шығару бағдарламалары арасындағы қатынас 2 принцип бойынша асырылады:

1) синхронды – іске қосылған енгізу-шығару операциясы аяқталмайынша, шақырған процесс тосқауылға алынады;

2) асинхронды – енгізу-шығару операциясы іске асырылғаннан кейін шақырылған процесс енгізу-шығару нәтижелері талан етілетін нүктеге дейін жұмыс істейді.

Яғни дайындық туралы сұрау режимінде сыртқы құрылғымен деректер айырбасы процесін басқарып отыратын драйвер «дайындық туралы сигналдың бар-жоқтығын тексеру» командасын циклде орындайды. Сигнал пайда болмайынша драйвер еш нәрсе істемейді. Осының арасында орталық процессордың уақыты рационалды емес түрде қолданылады. Енгізу-шығару командасын бергеннен кейін

енгізу-шығару құрылғысы туралы уақытша ұмытып, басқа бағдарламаның орындалуына көшкен жөн, ал дайындық туралы сигналды енгізу-шығару құрылғысынан шығу туралы сұраныс ретінде қарастыруға болады. Осы дайындық туралы сигналдар ұзу туралы сұраныс сигналдары деп аталады. Ұзу режимінде жұмыс істейтін драйверлер бағдарламалық модульдердің күрделі кешені болып табылады және олардың бірнеше секциялары болады: іске қосу секциясы, бір немесе бірнеше жалғастыру секциялары және аяқтау секциялары. Құрылғылар өзінің дайындығы туралы ақпаратты процессорге тікелей емес, арнайы ұзу контроллері арқылы береді. Мұнда ол процессормен байланыс жасау үшін, тек бір желіні ғана емес, толық ұзу шинасын да қолданады. Үзілу пайда болған кезде ұзу контроллері өзінің күйі регистріне жазып алатын әр құрылғылардың ұзу нөмері болады. Үзілу нөмері арнайы үзілу кестесінде индекстерінде жұмыс істейді. Бұл кесте есептеу жүйесінің инициализация кезінде бөлінген орында сақталып, ұзу векторлары деп аталағын үзілуді өңдейтін бағдарламалардың мекен-жайын сақтап отырады. Командалар орындалып жатқан кезде, процессор сыртқы үзуді байқайды. Процессор үзілуді өңдеуге көшкенде, өзінің күйінің бір бөлігін келесі команданы орындау үшін сақтап отырады. Үзілу процессордың жұмысымен бірге асинхронды орындалады және бағдарламалаушы үзілудің қай жерде пайда болатынын білмейді. Процессор командаларды орындап жатқан кезде ғана ерекше жағдайлар пайда болуы мүмкін. Мысалға, голып кету жағдайы, 0-ге бөлу, жадының жоқ бетіне үндеу, т.б. Мұндай ерекше жағдайларға келесілер тән: командалар орындалып жатқан кезде процессор ерекше жағдайларды табады; процессор ерекше жағдайды орындауға көшкенде, өзінің күйінің бір бөлігін ағымдағы команданы орындауға сақтап қалады; процессор жұмыс істеп жатқан кезде ерекше жағдайлар онымен бірге синхронды түрде пайда болады, бірақ ерекше жағдайлар бағдарламалаушылардың күтпейтін жерінде пайда болады.

Бағдарламалық үзілу жүйелік шақырулардың ішінде ерекше жұмыстарды істеу үшін арнайы командалар іске асырылып болғаннан кейін пайда болады. Бағдарламалық үзілулер келесі қасиеттерге ие:

1) бағдарламалық үзілулер арнайы командалардың орындалу нәтижесінде пайда болады;

2) процессор бағдарламалық үзілуді орындаған кезде өз күйінің бір бөлігін келесі команданы орындамас бұрын сақтап қалады;

3) бағдарламалық үзілулер процессордың жұмысымен синхронды пайда болады және бағдарламалаушылармен абсолютті түрде болжамды.

Үзілу режимінде енгізу-шығару операцияларын басқару бағдарламалаушылар үшін қиындау болады, себебі мұндай бағдарламаларды жасау күрделі. Мысалға, баспа драйверлері жағдайы. Windows операциялық жүйесінде параллельді порт арқылы баспаға шығару үзілу режимінде емес, баспа орындалып жатқан уақыттың толығында орталық процессордің жүктелуіне әкелетін дайындық туралы сұрау режимінде жұмыс істейді. Бұл уақытта іске қосылған басқа да жұмыстар іске асырыла береді. Бірақ аталған операциялық жүйелер ығыстырушы мультитапсырмалықты қолдайтындықтан басқа да жұмыстар іске асырылады.

Қарастырылған басқарудың екі принципі байт-бағдарлы сұлба немесе блок-бағдарлы сұлба бойынша енгізу-шығару ағындарын қалыптастыратын категориялары бар құрылғылар кластарына негізделген. Байт-бағдарлы сұлба құрылғылары байттық айырбас жүргізеді, ал блок-бағдарлы – байттар тобын.

Құрылғыны жүйеде баяндау үшін белгілі бір құрылғының нақты сипаттамасы және негізгі параметрлері бар құрылғыны басқаратын блок-сұлба болады.

Блок-бағдарлы және байт-бағдарлы құрылғыларға арналған басқару блоктарының ұқсастықтары мен айырмашылықтары:

1) белгілі бір категориялы құрылғыларға арналған кестенің жалпы өрістері мен элементтеріне: құрылғының аты; атрибуттары (сипаты); құрылғы нөмері; ауыстыру көрсеткіші; буферизация және жазба көлемі жатады. Блок-бағдарлы құрылғының қосымша элементтеріне: сұраныс өңдеу күйінің көрсеткіші; күтуші семафордың мекен-жайы; құрылғының үзіліссіз жұмыс істеу интервалы; жаңылу санауыштары жатады;

2) құрылғы аты – операциялық жүйедегі құрылғының символдық көрсетілуі. Құрылғының нөмерімен бірге оның аты құрылғының жүйедегі физикалық мекен-жайын анықтайды;

3) атрибуттар – биттердің күйін көрсетіп тұратын екілік түрді басқарушылардың қондысы. Атрибуттардың екі биті тікелей мүмкіндікті құрылғының жағдайын анықтайды, екі бит енгізу-шығару бойынша буферизацияның мәнін анықтайды, келесі екі биттер – жазу және оқу бойынша қорғаныс, екі бит – тікелей мүмкіндікті құрылғының қағынасуға дайындық көрсеткіші;

4) буферизация және жазба өлшемдері құрылғының физикалық ерекшеліктерімен анықталады;

5) құрылғының үзіліссіз жұмыс істеу интервалы – тайм –аут механизмі бойынша атқарылатын жұмыстың басқарылуын қамтамасыз етеді;

б) бөлінбейтін енгізу-шығару құрылғыларын бір уақытта қолдану үшін снулинг қосылады. Снулингтің негізгі мақсаты – енгізу-шығару құрылғыларының бір-бірінен бөлініп тұрғанын көрсету.

Ор операциялық жүйе енгізу-шығарудың буферизациясын қамтамасыз етеді. Дискілік кэштегі деректерді буферизациялау немесе кэштеуді негізгі ішкі жүйелердің өнімділігіндегі қарама-қайшылықтарды жою үшін пайдаланады. Деректерді оқудың дискілік операциясын жылдамдаттыру үшін екілік буферизация қолданылады. Мұнда бірінші буферге магниттік дискідегі деректер енгізіліп жатса, екінші буфердегі есептелген деректер орындалып жатқан тапсырмада оқылып, өрі қарай берілуі мүмкін.

Дискілік кэш (кэш буфері) – жүйелік процессормен басқарылатын буферлер жиынтығы. Белгілі бір файлдың жазбалары бар секторлар жиыны есептеліп, кэштен өткенде сол жерде қалады (осы буферлерді басқа секторлар ауыстармағанша). Егер деректі қайтадан оқу керек болса, онда оны жедел жадыдан дискімен байланыспай-ақ алуға болады.

Негізгі әдебиет: [1, б. 247-278], [2, б. 158-182]. Қосымша әдебиет: [9, б. 269-300].

Бақылау сұрақтары

1. Дискінің бірінші секторының құрамын баяндаңыз.
2. Partition Table не үшін керек?
3. Магниттік дискіде қанша және қандай бөлімдер болуы мүмкін?
4. Компьютерді қосқаннан кейін операциялық жүйенің жүктелуі қалай жүзеге асады?
5. Енгізу-шығарумен басқару жүйесінің мақсаты, функциялары.
6. Есептеу жүйесінде енгізу-шығарумен басқарудың принциптері.
7. Синхронды және асинхронды енгізу – шығару айырмашылықтары.
8. Дискілік КЭШ жұмысының ұйымдастырылуы.

4 Операциялық орта

4.1 Есептеу процесі және процестерде жүргізілетін операциялар ресурсы

Операциялық жүйе есептеу процестерін басқарып, процестер

арасында ресурстарды үлестіреді және тұтынушылардың қолданбалы бағдарламасы іске асатын бағдарламалық ортаны құрайды.

Операциялық орта – бағдарламаларға және қолданушылардың белгілі бір сервисті алу үшін операциялық жүйеге кіру үшін керекті интерфейсдер.

Операциялық жүйе бірнеше операциялық ортадан тұрады.

Операциялық орта бірнеше интерфейсден тұруы мүмкін: қолданушылық және бағдарламалық. Операциялық орта бағдарламалық интерфейсдермен анықталады. Қолданбалы бағдарламалар интерфейсінә процесстерді, жадыны, енгізу-шығарумен басқару жатады.

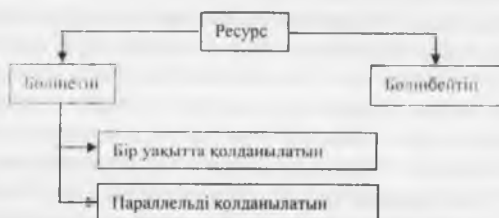
Тізбекті процесс – бағдарламаның деректерімен оның тізбекті процессорде орындалуы. Процесс тұжырымдамасы екі аспектіден тұрады: біріншіден, ол деректерді тасымалдаушы болып табылады, екіншіден, ол деректерді өңдеумен байланысты операцияларды ози орындайды.

Процестерге мысал ретінде қолданушылардың қолданбалы бағдарламаларын, утилиттерді, басқа жүйелік өңдеу бағдарламаларын қарастыруға болады. Текстігі түзету, бастапқы бағдарламаны көрсету, орындау, байланыстыру процеске жатады. Мұндағы бастанқы бағдарламаны көрсету бір процесс, ал келесі бастанқы бағдарламаны көрсету екінші процесс болады. Себебі транслятор бағдарламалық модульдерді біріктіруінә есебінде бір бағдарлама болса, оны өңдеп отырған деректері әр түрлі. Операциялық жүйені қарастырғанда оның негізгі параметрлері – ресурс және процессті қарастырамыз.

Ресурс – қайталана қолданылатын, салыстырмалы түрде тұрақты, жиі жетіспейтін объект. Бұл объектілер екінші (активті) кезде процесстермен шақырылып, қолданылып, босатылады. Ресурстар бөлінетін және бөлінбейтін болады. Бөлінетін ресурстарды бірнеше процесстер бір уақытта немесе параллельді қолданады.

Алғашқы жүйелерді жасағанда ресурстар деп процессорлық уақыт, жады, енгізу-шығару каналдары және шалғай құрылғылар (периферийные устройства) саналды. Бірақ аз уақыт ішінде ресурстар ұғымы амбебап және жалпы мәнге ие бола бастады. Түрлі бағдарламалық және ақпараттық ресурстар жүйе үшін объект ретінде қарастырылады. Мүндай объектілер бөлініп, үлестіріле алады және оған қатынас жасау мүмкіндікті сәйкес түрде бақылап отару керек. Қазіргі кезде ресурс өзінні мағынасы атрибуттарға толы абстрактілік структураға айналған. Атрибуттар құрылымға қатынас жасау мүмкіндігін және оның жүйедегі физикалық мәнін сипаттайды. Жүйелік ресурстардан бөтен хабар және синхросигналдар түрінде

болатын объектілер де ресурстар деп атала басталды.



4.1 Сурет – Ресурстың бөліну блок-сұлбасы

Есептеуінің машина құрамына арнайы контроллердің енгізілуі алынған деректерді шығару және орталық процессорда келесі есептеулерді жүргізу процестерін біріктіруге мүмкіндік берді. Бірақ та процессордың енгізу-шығару операциясының аяқталуын күтуі ұзақ және жиі бола берді. Сондықтан да есептеу жүйесінің мультибағдарламалық немесе мультитапсырмалық режимін ұйымдастыру қолға алынды.

Мультибағдарламалық режимнің басты мақсаты – бірінші бағдарлама енгізу-шығару операциясының аяқталуын күтпн отыра, екінші бағдарлама қарастырылу үшін дайын тұрады. Осы арқылы бірнеше тапсырмалар жиынын орындауға кететін уақытты үнемдеп, бар ресурстарды қолдануға мүмкіндік туады.

Ресурстардың (негізгі ресурстар – процессор, жады, енгізу-шығару құрылғысы) маңызды түрінің бірі – бағдарламалық модульдер. Ең бастысы – жүйелік бағдарламалық модульдер, себебі олар бағдарламалық ресурстар саналып, орындалып жатқан процестер арасында үлестірілуі мүмкін.

Бағдарламалық модульдер бір ретті қолданбалы және бірнеше ретті (қайталана) қолданбалы болуы мүмкін. Бір ретті қолданбалы деп бір рет ғана қолданылатын бағдарламалық модульдерді айтамыз, яғни олар орындалып жатқан кезде өздерін бұзуы мүмкін: есептеу қадамына тікелей байланысты кодтың бір бөлігі немесе берілген деректер зақымдануы мүмкін. Бір ретті қолданбалы бағдарламалық модульдер бөлінбейтін ресурстар болып табылады. Оларды тіпті жүйе ресурстары ретінде қарастырмайды. Бір ретті қолданбалы бағдарламалық модульдер операциялық жүйе жүктеліп жатқан кезде іске қосылады. Ескере кететін факт: осы модульдер жазылған екілік файлдар бұзылмайды, яғни операциялық жүйенің келесі қосылуында тағы да қолданылуы мүмкін.

Қайталана қолданбалы бағдарламалық модульдер ерекше артықшылығы жоқ (непривилегированный), ерекше артықшылығы бар. ресентерабельді болуы мүмкін.

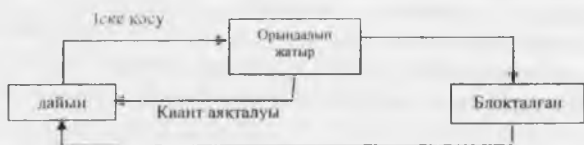
Ерекше артықшылығы бар бағдарламалық модульдер ерекше артықшылықты деп аталатын режимде жұмыс істейді, яғни ешқандай сыртқы жағдай табиғи есептеу ретін бұзбай, үзілу жүйесі сөндірулі болған кезде іске қосылады. Нәтиже секілді бағдарламалық модуль аяғына дейін орындалады, одан кейін басқа есептеу процесінен шақырылуы мүмкін. Сыртқы бақылаушыға есептеу процесі шақыратын ерекше артықшылығы бар бағдарламалық модуль бөлінетін ресурс болып көрінеді.

Ерекше артықшылығы жоқ бағдарламалық модульдер – жұмыс істеп тұрған кезде бөлішін кете алатын қаранайым бағдарламалық модульдер. Осыған байланысты оларды бөлінетін деп атауға болмайды, себебі есептелу процесінің жақтауында орындалып жатқан модульдің үзілуінен кейін басқа есептелу процесінің талабы бойынша қайта қосылмайды.

Оған қарама-қарсы ресентерабельді бағдарламалық модульдер оның орындалуының бірнеше рет үзілуін және қайта іске қосылуын қамтамасыз етіп тұрады. Ол үшін ресентерабельді бағдарламалық модульдер үзілмелі есептеулер үшін арналған аралық нәтижелердің сақтығы және есептеу процесі үзілген жерінен жұмысын қайта бастаған кезде аралық нәтижелерге қайта оралу мүмкін болатындай етіп ұйымдастырылуы керек.

Ресентерабельді бағдарламалық модульден басқа қайта кірмелі де бар. Ол да оның бірнеше ретті параллельді қолданылуын қамтамасыз етеді, бірақ ресентерабельді секілді үзуге болмайды.

Ақпараттық ресурстар да бар, яғни ресурс қызметін деректер атқарады. Олар жедел жадыдағы айнымалылар түрінде немесе файлдар түрінде болуы мүмкін. Егер де процесстер деректерді оқу үшін ғана қолданса, онда ақпараттық ресурстарды болуға болады. Ал егер процесстер ақпараттық ресурстарды өзгертсе, онда мұндай деректермен жұмысты арнайы түрде ұйымдастыру керек.



4.2 Сурет -- Процесс күйінің графы

Жүйеде пайда болатын әрбір процесс дайындық күйіне түседі.

Операциялық жүйе қайсыбір жобалау алгоритмімен қолдана осы дайын процестердің біреуін таңдап, оны орындалу күйіне аударады. Орындалу күйінде процестің програмалау кодының тікелей орындалуы жүреді. Мұндай күйден шығу үш себептен орындалады:

- 1) операциялық жүйе процестің орындалуын қолдамайды;
- 2) белгілі бір оқиға пайда болмайынша процесс өзінің жұмысын жалғастыра алмайды және операциялық жүйе оны блок күйіне өкеледі;

- 3) есептеу жүйесінде үзілудің пайда болу нәтижесінде оны дайындық күйіне ауыстырады.

Белгілі операциялық жүйелерде процестің жағдайы көбірек детализданған болып, бір жағдайдан екінші жағдайға өтудің жаңа варианттары пайда болуы мүмкін. Мысалға, Windows NT операциялық жүйесіне арналған процестің күйі Unix-9 операциялық жүйесіне арналған 7 түрлі күйге ие.

Бірақ барлық операциялық жүйе жоғарыда аталған модельдерге тәуелді.

Process Control Block (PCB) – дескриптор – операциялық жүйе процесс туралы барлық маңызды ақпаратты сақтайтын объект.

Жалпы жағдайда дескриптор келесі ақпаратқа ие:

- 1) процесс идентификаторы;
- 2) супервизорға берілген ресурстардың ережелерін анықтайтын тұлға немесе класс;
- 3) процесс приоритеті;
- 4) процесс айналымы (дайын, блокталған, орындалу);
- 5) процессор регистрінің ағынды мәні сақталған және қорғалған жады облысы;
- 6) ресурстар туралы ақпарат;
- 7) басқа процесспен байланыс жасауды ұйымдастыратын орын;
- 8) іске қосу уақытының параметрі;
- 9) жүйеде басқару файлдар жоқ болған кезде мекен-жай және дискідегі тапсырмалар туралы ақпарат.

Сипаттаушылар, дескрипторлар жедел жадыда орналасқан. Олар операциялық жүйенің жұмысын процессорды ақпараттық құрылыммен аппараттық қолдау үшін сәйкесті механизмдер арқылы жүзеге асырады, себебі 180*86-да бір тапсырмадан екінші тапсырмаға өткенде процессор регистрларының құрамы сақталатын TSS тапсырмалар күйінің сегментінің орналасу жерін көрсететін TR (Task Register) регистрі бар. Қазіргі операциялық жүйедегі микропроцессорларға арналған тапсырмалар дескрипторында TSS бар.

Процесті басқарудың блогы сақтау үшін арналған ақпаратты әрі қарай қарастыру үшін екі бөлікке бөлген дұрыс. Процессор регистрінің құрамын (бағдарламалық санауыш мәнін қосқанда) процестің регистрлік контексті деп, ал қалғанын – процестің жүйелік контексті атаймыз. Процестің регистрлік және жүйелік контекстін білу операциялар орындау арқылы оның операциялық жүйедегі жұмысын басқаруға мүмкіндік береді. Бірақ бұл процесті сипаттау үшін жеткіліксіз. Операциялық жүйеге процестің қандай есеңтеулермен айналысып жатқанына бәрібір, яғни қандай код және қандай деректер оның мекен-жай кеңістігінде орналасқаны қызықтырмайды. Қолданушының көз қарасы бойынша керісінше, процестің мекен-жай кеңістігінің құрамы көбірек қызығушылық тудырады. Процестің мекен-жай кеңістігінде орналасқан код және деректер қолданушы контекст деп аталады. Процестің регистрлік, жүйелік және қолданушы контекстердің қосындысын қысқарту үшін жай процестің контексті деп аталады. Уақыттың әр кезінде процесс өзінің контекстімен сипатталады.

Процестермен салыстырғанда:

1) мультибағдарламалық идеяның іске асуы үшін процесс ұғымы енгізілген;

2) мультитапсырмалар үшін ағын (тредтер, жіптер) ұғымы;

3) ағындар процестер ішкі параллелизмді қолдану үшін жұмыс істейді (контасырмалық қолданбалы бағдарламада бірнеше операцияларда параллельді түрде орындау мүмкіндігін туғызады).

Әр процесске өзінің ресурстары тағайындалады: файлдар, терезелер, семафорлар және т.б, ал барлық тредалардың сол өзінің процестерінің виртуалды кеңістіктері бар, яғни глобалды айнымалыларды ажыратады.

Процестерге операциялар қолдану. Есеңтеуші жүйенің тиімділігіне жадыны үлестіру және орындаушы процестер арасындағы процессорлы уақыт әдістері әсер етеді. Ресурстарды бөлу туралы айтқанда, (термин тапсырма ағын түсінігін қамтиды) операцияны жүйе процесті басқарумен байланысты келесі функцияларды орындайды:

1) процесті ойлап табу және өшіру;

2) процестерді жоспарлау және диспетчеризациялау;

3) тапсырмалардың синхронизациясы, оларды коммуникация құралдарымен қамтамасыз ету.

Процесті ойлап табу келесі операциялардан тұрады:

1) процесске атын меншіктеу;

2) сол атауды жүйеге белгілі атаулар тізіміне қосу;

- 3) процесстің бастапқы басымдылығын анықтау;
- 4) басқару блогын РСВ процесімен ұйымдастыру (Талсырманы жасау және оңіру колданушылардың немесе тапсырмалардың сәйкес сұраулары арқылы жүзеге асырылады).

Процесс жаңа процессті туғызуы мүмкін. Бұл жағдайда тудырушы процесс аталық процесс деп, ал ойлап табылған процесс туынды деп аталады. Мультибағдарламалық операционды жүйенің қызметі әр түрлі процесстердің үстінде орындалатын операциялар тізімінен тұрады және процессті бірінен екіншісіне ауыстыру онымен қатар жүреді.

Процессормен кейбір процессті орындау барысында құрылғыда операцияның біткендігін көрсетін отыратын енгізу-шығару құрылғысынан үзілу болып тұрады. Орындалатын процессте сәл тоқтату операциясы жүргізіледі. Әрі қарай операционды жүйе енгізу-шығаруға сұрауды инициалдаған процессті блоктан шығарады, жоспарды орындау кезінде жаңа таңдалған немесе тоқтатылған процессті қосуды жүзеге асырады. Біз көріп отырғанымыздай, енгізу-шығару операциясының аяқталуы туралы мағлұматты өңдеу нәтижесінде орындалып отырған процесстің ауысуы болуы мүмкін. Процессорды бір процесстен басқасына дұрыс ауыстыру үшін орындалып жатқан процесстің контекстің сақтап қалу қажет және процессорға қосылатын процесстің контекстің қалпына келтіру керек. Процесстердің жұмыс істеу қабілеттілігінің осындай сақтау/қалпына келтіру процедурасы контекстік ауыстыру деп аталады.

Негізгі әдебиеттер: [1, б. 16-49], [2, б. 75-91], [3, б. 139-190], [4, б. 145-176]. Қосымша әдебиет: [9, б. 31-49].

Бақылау сұрақтары

1. «Ресур» түсінігін анықтаңыздар.
2. Ресуретардың қандай типтері мен түрлерін білесіздер?
3. Мультибаба дарламалық режимнің түр-негізі неде?
4. Ерекше артықшылығы бар бағдарламалық модуль дегеніміз не?
5. Процесс дегеніміз не?
6. Процесс қандай негізгі жағдайларда болуы мүмкін?
7. Процессті басқару блогының негізгі құраушылары анықтаңыздар.
8. Ағын дегеніміз не?
9. Процесстерді басқарумен байланысты негізгі функцияларды

атаңыз.

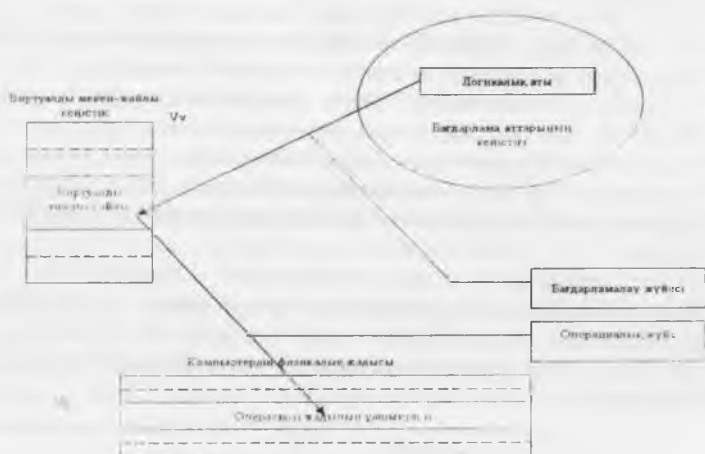
10. Процестің контексті деген не, оның құрылымы қандай?
11. Контексті ауыстыру қашан болады?

5 Жадымен басқару

5.1 Жады және бейнелер, виртуалды адрестік кеңістік

Айнымалылардың және бағдарламалық модульдердің кіріс нүктелерін атаулар кеңістігі құрайды. Физикалық жады – ол ұяшықтардың (шектелген) реттелген жиыны.

Қамтамасыз етілген жүйелік бағдарлама қолданушы белгіленген атауды физикалық ұяшықтың жадысымен байланыстыру керек, яғни атаулар кеңістігін компьютердің физикалық жадысына көрсету керек. Жалпы жағдайда бұл бейне 2 кезеңде жүзеге асырылады: алдымен бағдарламалау жүйесімен, ал кейін операционды жүйесмен.



5.1 Сурет – Виртуалды мекен-жайлы кеңістік блок-сұлбасы

Виртуалды мекен-жайы – бағдарламалау жүйесімен орындалуға тағайындалған дайын машиналық командалар мен айнымалылардың бағдарламадағы мекен-жайы.

Екінші бейне жылжытушы жүктеушінің көмегімен жүзеге асырылады. Бағдарламаны жүктегеннен кейін виртуалды адрес жоғалып кетеді, ал мүмкіндік тікелей физикалық ұяшықтарға

орындалады.

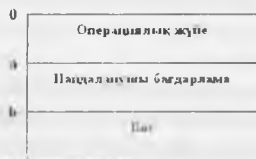
Мынадай орын алады:

$$V_v < V_p$$

$$V_v = V_p$$

$$V_v > V_p$$

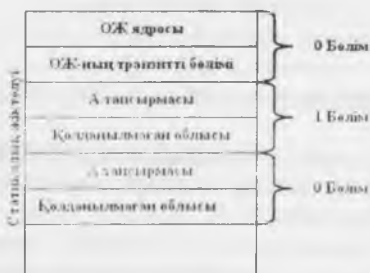
Жадыны ұйымдастыру – ол негізгі жадыны көрсету және қолдану тәсілі. Жадыны ұйымдастырудың негізгі тәсілдері.



5.2 Сурет – Жай үздіксіз (бір бағдарламалық жүйелер үшін) блок-сұлбасы

Мысалы – MS-DOS. Логикалық және виртуалды адрес кеңістігі жадының бос ауданынан немесе оперативті жадының бүкіл мүмкін көлемінен үлкен болуы қажет бағдарламаны ойлан табу үшін асырумен орналастыру – оверлейлік құрылым қолданылады. Өрбір оверлейлік бағдарламада бір негізгі бөлім main және бірнеше сегменттер бар және машинаның жадысында бір мезгілде тек оның негізгі бөлімі мен бір немесе бірнеше жабылмайтын сегменттер болуы мүмкін.

Статикалық және динамикалық бөлімдермен үлестіру мультибағдарламалық режимдерде қолданылады. Операциянды жүйе ядросы жоқ жады бірнеше үздіксіз бөлімдерге бөлінуі мүмкін: бекітілген-статикалық, динамикалық – жаңа тапсырма пайда болғанда көрінеді.



5.3 Сурет – Статикалық жіктелу

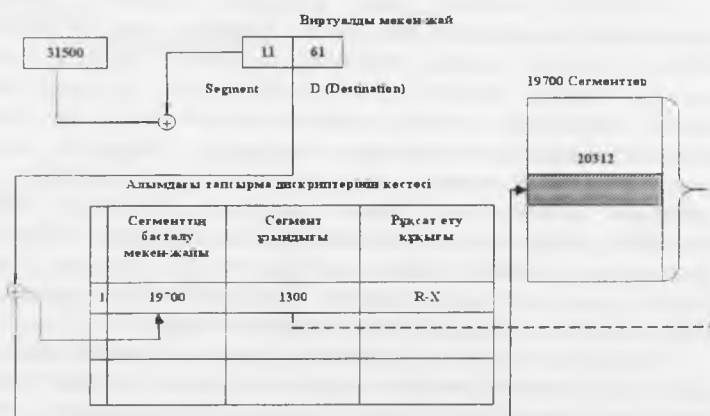
Жадының көлемі үлкен болмаған жағдайда параллель орындалатын қосымшалардың санын Swapping арқылы көбейтуге болады. Свопинг кезінде тапсырма түгелімен магнитті дискіге жүктелуі (сыртқы жадыға ауыстырылуы) мүмкін, ал оның орнына дискте тоқтатылып тұрған ерекше артықшылығы бар немесе орындалуға дайын жай тапсырма жүктеледі. Свопингтің жадыны басқаруға тікелей қатысы жоқ, ол процестерді жоспарлау ішкі жүйесімен байланысты. Свопинг контексті ауыстыру уақытын үлкейтеді. Дисктегі арнайы бөлінген кеңістігін (свопинг үшін бөлім) ұйымдастыру әсерінен жүктен шығару уақыты қысқаруы мүмкін. Дискпен ауыстыру үлкен өлшемді блоктармен жүзеге асырылады, яғни стандартты файлдық жүйеге қарағанда тезірек.

Оверлейлік құрылым. Процестің логикалық адрестік кеңістігінің өлшемі оған бөлінген бөлімнің өлшемінен (немесе ең үлкен бөлімнің өлшемінен) үлкен болуы мүмкін, сондықтан кейде оверлей (overlay) деп аталатын техника немесе асып түсетін құрылымды ұйымдастыру қолданылады. Негізгі идеясы – дәл берілген уақытта қажет болатын бағдарламаның нұсқауларын жадыда ұстау. Егер жүйенің логикалық адрестік кеңістігі аз болса, жүктеудің қажеттігі пайда болады, мысалы: 1 Мбайт (MS-DOS) немесе 64 Кбайт (PDP-11), ал бағдарлама оған қатысты биік. Виртуалды адрестік кеңістігі гигабайтпен өлшенетін жаңа 32-бөлімдік жүйелерде жадының жетіспеу мәселелері басқа әдістермен шешіледі. Бағдарламаның оверлейлік құрылымының бұтақтарының кодтары дискте жадының абсолютті кейсінде болады және керек болса, оверлейлер драйверлерімен есентеледі. Жай файлдық құрылымы бар жүйелерде оверлейлер түгелімен қолданушылар деңгейінде жүзеге асырылуы мүмкін. Оверлейлік құрылым (ОЖ) бірнеше ғана енгізу-шығару операцияларын орындайды. Типтік шешуі – бағдарламаның асып түсетін бұтақтарына қатынасу керек болған жағдайда жүктеуші қосатын арнайы командаларды линкермен туындылау.

5.2 Жадымен басқару. Сегменттік, беттік және сегментті-беттік жадыны ұйымдастыру

Сегменттік, беттік және сегментті-беттік жадыны ұйымдастыру жадыны орналастырудың үзілістік әдісіне жатады.

Бекітілген өлшемнің блогы – беттер, айнымалы өлшемнің блогы – сегменттер. Виртуалды адресстер нақты адресстерге блоктарды көрсететін кестелер көмегімен айналады.



5.4 Сурет – Виртуалды жадыны сегменттік тәсіл арқылы ұйымдастыру

Операциялық жүйе (ОЖ) әрбір колданылатын процесс үшін сегменттердің дескрипторларының сәйкес кестесін құрады. Егер тапсырма сегменті берілген уақытта ОЗУ-да тұрған болса, ол биттің ішінде бояумен белгіленеді. Бұл жағдайда жады диспетчері «адрес» өрісіне физикалық жады адресін жазады, ал «сегмент ұзындығы» өрісіне адрестелетін жады ұяшықтарының санын жазады. Егер present биті қазір сегмент сыртқы жадыда орналасқандығын көрсетсе, аталған өрістер сыртқы жадыдағы сегмент адресін көрсетеді. Сегмент дескрипторында сонымен қатар оның типі туралы (кодтың сегменті немесе мәліметтер сегменті) мәліметтер, осы сегментке кіру құқы бар. Басқаруды басқа тапсырмаға берген кезде ОЖ осы тапсырманың сегменттерінің дескрипторларының кестесінің адресін сәйкес регистрге енгізуі тиіс.

Мәселелерді шешу үшін ауыстыру диспетчері (не сыртқы жадыға ауыстырылған, не жаңа сегментпен ауыстырылған сегментті анықтау керек) келесі дисциплиналарды қолданады:

- 1) FIFO (First In First Out);
- 2) LRU (Last Recently Used) – соңғысы жуықта ғана қолданылған, көбінесе қолданылмаған;
- 3) LFU (Last Frequently Used) – басқаларынан аз қолданылатын – Random – кездейсоқ.

Виртуалды жадыны ұйымдастырудың беттік әдісінде жады да,

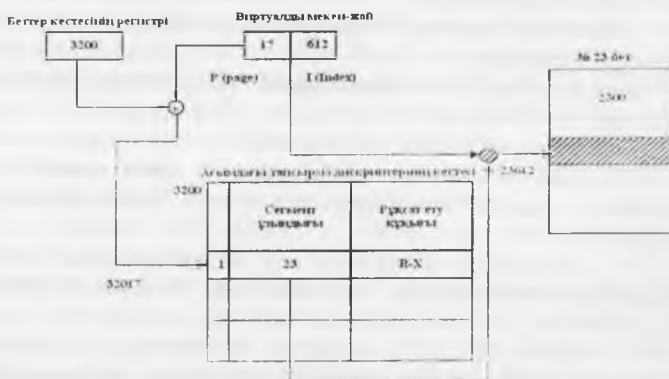
бағдарлама да бірдей бөлімдерге, беттерге бөлінеді. Тапсырманың виртуалды беттер бөлімі оперативтік жадыда, ал басқа бөлімі сыртқы жадыда (SWAP-файл немесе беттік файл) орналасады. Кейбір ОЖ-ларда жүктеп шығарылған беттер файлда емес, дисктік кеңістіктің арнайы бөлімінде орналасады. UNIX жүйелерінде осы мақсатпен арнайы бөлімдер бөлінеді. Физикалық адрес жүнінен (P_r, i), ал виртуалды - (P_v, i) жіберіледі.

P_v – виртуалды беттегі нөмір.

P_r – физикалық беттегі нөмір

i – бет ішіндегі ұяшық индексі

Бет дескрипторының сегмент дескрипторынан айырмашылығы оның ұзындық өрісі жоқ. Беттік жадыны қорғау сегменттік механизм жағдайындағыдай әрбір бетке мүмкіндік деңгейінің контроліне негізделген.



5.5 Сурет – Виртуалды жадыны ұйымдастырудың беттік тәсілінің сұлбасы

Жаңа ОЖ-да көбінесе LRU беттерін ауыстыру дисциплинасы ең тиімдісі ретінде қолданылады.

Сүйреу – керек бетті жүктеу активті жұмыс отыр жатқан бетті сыртқы жадысына ауыстыруды шақырғандағы жағдай. Сүйреумен күресу үшін «жұмыс істейтін жиындар әдісі» пайдаланылады. Жұмыс істеуші жиын – T интервалында активті беттердің жиыны.

Егер ОЖ орындалын жатқан тапсырмалардың жұмыс істеуші жиынын анықтай алатын болса, онда сүйреуді қайтару үшін тек жұмыс істеуші жиындарының қосындысы жүйенің мүмкіндігінен аспайтындай тапсырмаларды орындауды жоспарлау жеткілікті.

Есептеуіш машинаның жұмысын тездету үшін виртуалды жадының ұйымдастырудың сегменттік және беттік әдістері үшін дескрипторларды бүркемелеу қолданылады. Бүркемелеудің ең тиімді тәсілі ассоциативтік бүркеме.

Ассоциативтік жадтаушы құрылғының әркеті ету принципі. Жадның ұяшығының құрамын идентификациялауға мүмкіндік беретін кілт (адрес) жазылатын ұяшық сәйкес жадның әрбір ұяшығына қойылады. Ақпараты бар, қалған мәліметтерді идентификациялауға мүмкіндік беретін сәйкес ұяшықты әдетте тег өрісі деп атайды. Барлық жадылардың өріс тегтерін, жадның ассоциативтік құрылғысын қарап шығу бір мезгілде жүзеге асады, яғни тегтің әрбір ұяшығында биттік конъюнкция құралымен жадыға бір қатынағанда мәліметтерді олардың белгілері бойынша табуға мүмкіндік беретін қажетті логика бар. Тег өрісі аргумент, деректер өрісі функция деп аталады.

Виртуалды жадының ұйымдастырудың сегменттік және беттік тәсілдерін салыстыру

Беттік тәсілдің негізгі артықшылығы – минималды мүмкіндікті ұшылдәу.

Кемшіліктері:

1) беттердің кестесін орналастыру үшін үлкен көлемді оперативтік немесе бүркемелік жады және өндеу уақыты талап етіледі;

2) беттерді бөлу кодтарда бар логикалық ара-қатынастарсыз іетеледі, бетаралық өтулер сегментаралық өтулерге қарағанда жиі кездеседі.

5.1 Кесте – PC процессорлары қатынай алатын нақты максималды және виртуалды жады

Процессорлар	Нақты жады	Виртуалды жады
8088	1М	-
8086	1М	-
286	16М	11
386	4Г	64Г
---	---	---
486	4Г	64Г
Pentium	4Г	64Г

5.3 Жадымен басқару. Жаңа компьютерлерде жадының көп түрлілігі

1) Conventional Memory;

2) Upper Memory Area (UMA) - жоғарғы жадының аймағы;

- 3) High Memory Area (HMA) - үлкен жадының аймағы;
- 4) Extended Memory – кеңейтілген жады;
- 5) Expanded Memory – қосымша жады, бейнеленетін жады;
- 6) Video Ram Memory – бейнежад;
- 7) Адаптерлердің ПЗУ және арнайы ОЗУ;
- 8) BIOS жүйесімен ПЗУ.

Intel 80 сериялы процессорларда қолданылатын жадының сегменттік ұйымдастыруының механизмі келесідей: физикалық адрес екі компоненттен алынады – сегменттік адрестен және ығыстырудан. Әрбір компонент екі байттық. Процессордың нақты жұмыс істеу режимінде солға қарай төрт битке жылжытылған сегменттік компонентке адрес есептеу үшін ығысу компоненті қосылады. Мысалы: 1234: 0005-сегментті адрес 1234, ығысу 0005. физикалық адрес: $12340 + 0005 = 12345$.

Негізгі жады. Бірінші PC және XT компьютерлерінде жадының жұмыс істеуінің кеңістігі 1 М камтиды. Бұл кеңістік бөлімдері арнайы мақсаттарға арналған облыстарға бөлінген. Бағдарламалар негізгі жады деп аталатын ОЗУ аймағына жүктеледі. Оның бірінші PC өлшемі 512 К. Қалған 512 К ішкі қолдану үшін қосалқыланды. Кейінгі компьютерлерде көлем 640 К-ға дейін үлкейтілді.

Жоғарғы жады аймағы (UMA) – диапазоны 640 Кбайттан 1 Мбайтқа дейін. UMA бірнеше бөлімдерге бөлінеді:

1) бірінші 128 Кбайт – бейнежад, бейнеадаптерлердің қолданылуына арналған;

2) келесі 128 Кбайт – ПЗУ микросхемаларында сөйкес үлкейту слоттарында орнатылған тақшаларында жазылған BIOS адаптерлік бағдарламаларға арналған. VGA мен олармен үйлесімді бейнеадаптерлері өздерінің BIOS-тары үшін осы аймақтың 32 Кбайтын қолданады;

3) соңғы 128 Кбайт жүйелік BIOS үшін. Осы аймақта өзін-өзі тексеру бағдарламасы және бастапқы жүйелік жүктеуші сақталады.

Кеңейтілген жады. 286 процессоры бар компьютердің жады картасы 1 Мбайттан асады. 286 және 386 SX компьютерлерінде кеңейтілген жады 1 Мбайттан 16 Мбайтқа дейінгі көлемді камтиды, ал 386DX, 486 және Pentium – компьютерлерінде 4Гбайт. Осы жадыға қатынау процессор қорғау режимінде тұрғанда ғана мүмкін.

XMS жады – кеңейтілген жадының спецификациясы (1988 жылы енгізілген), бағдарламаларға кеңейтілген жадыға қатынауға мүмкіндік беретін тәсілді анықтайды. 286 және одан жоғары процессоры бар компьютерлерге арналған және DOS режимінде жұмыс істейтін бағдарламаларға кеңейтілген жадыны қолдануға

мүмкіндік береді. XMS спецификациясымен сәйкес кеңейтілген жадыны ұйымдастыру үшін HIMEM.SYS қолданылады. HMA - High Memory Area — үлкен жады аймағы — бұл жадының 64 Кбайтқа қарағанда аз 16 байтты қамтитын және кеңейтілген жадының 1 Мбайттың басында орналасқан бөлігі.

Қосымша (көрінетін) жады. Expanded Memory — EMS-тың өргүрілігі.

Негізгі және кеңейтілген жадыға қарағанда процессор қосымша жадыны адрестей алмайды. Оған тек кішкентай жадының жоғарғы аймағында ұйымдасатын терезе арқылы қатынауға болады (64 Кбайт). EMS құралы деректерді қорғау үшін жадыны қамтамасыз етеді.

Оперативті жадыны қолдануды тиімдеу тез қозғалуды үлкейту немесе босату мақсатымен жүргізіледі Conventional memory:

1) тез өсер етуді жүзеге асыру;

2) жүзеге асады:

а) 8-16 разрядты ПЗУ-дің «жай» микросхемаларының құрамын тез қозғалушы 32-разрядты негізгі жадыға көшірумен түсіндірілетін көлеңкелі жадыны құрумен;

б) буферизациямен;

с) дисктік жадыны бүркемелеумен;

д) электронды дискті жасаумен;

е) босату Conventional Memory.

Жүзеге асады:

а) жоғарғы бос жадыны UMB - Upper Memory Block — жоғарғы жадының блогы түрінде қолданумен;

б) бірінші 64 Кбайтын кеңейтілген жадының қолданумен (HIMEM.SYS драйвері осы жадыны XMS спецификациясына сәйкес қолдануға мүмкіндік береді).

Буферлеу — оперативтік жадыдағы ішкі буферге файл үлкен өлшемді блоктармен көшіріледі, соңына буфер құрамы шығушы файлга жазылады.

Дисктік жадыны бүркемелеу — оперативті жады аймағында диск секторының құрамын — бүркеме буферін сақтау үшін кеңістік бөлінеді. Басында түгел осы аймақ бос болады. Бағдарлама дискпен жұмыс істей бастағанда, ол талап еткен секторлар бүркеме буферіне көшіріледі.

Егер бағдарламаға сектор керек болса, бүркеме драйвері оның бүркеме буферінде жоқ екендігін көрсетеді. Бар болса, дискті физикалық оқу орындалмайды. Бағдарлама буфердегі секторлардың көшірмесін қолданады. Егер талап етілген сектор бүркеме буферінде жоқ болса, ол дискіден оқылып бүркеме буферіне жазылады, ережеге

сойкес ескі бүркеме буферінің орнына дискті бүркемелеу тез қозғалтуды үлкейтуді береді.

Мысалы: FAT және каталогка байланыс жиі жүреді.

Дисктің физикалық бүркемесі әдетте кеңейтілген немесе қосымша жадыны қолданумен жүзеге асады.

Электронды диск – электронды дискіге жазылған деректер оперативті жадыда сақталады. Жай дисктен ол тез қозғалғыштығымен ерекшеленеді.

Негізгі әдебиет: [1, б. 72-79], [2, б. 221-238], [3, б. 357-378], [4, б. 310-322]. Қосымша әдебиет: [9, б. 72-79], [10, б. 160-173], [13, б. 192-224].

Бақылау сұрақтары

1. «Виртуалды адрес», «виртуалды адрестік кеңістік» деген не?
2. MS-DOS-та жадыны орналастыру әдістерін айтыңыз.
3. MS-DOS-та оверлейлерді қолдану нені білдіреді?
4. Свопинг деген не?
5. Виртуалды жадыны ұйымдастырудың сегменттік тәсілін түсіндіріңіз.
6. Виртуалды жадының сегменттік және беттік ұйымдасу тәсілдерін салыстырыңыз.
7. Беттерді ауыстыру тансырмаларын есептеуге қандай дисциплиналар қолданылады?
8. Сүйрелеу қандай жағдайларда пайда болады?
9. Нақты режимде физикалық адрес қалай анықталады?
10. Conventional Memory дегеніміз не?
11. Электронды дисктің артықшылығы мен кемшілігін атаңыз.
12. Көлеңкелі жады дегеніміз не?

6 ОЖ архитектурасы және қолданбалы бағдарламалаудың интерфейсі

ОЖ құрудың негізгі принциптері:

1) модульдік принцип. Ауыса алатын мүмкіндігі бар және қайтадан кіру мүмкіндігі бар модульдардың болуы (ресентабельді). (Ресентабельді модульдер есептеу жүйесінің ресурстарын үнемді түрде қолдануына мүмкіндік туғызады);

2) функционалды таңдау принципі. Үнемі оперативтік жадыда болатын маңызды модульдердің бөлігі ядро деп аталады. (Ядроның

қарнайым құрамы: үзіліс жүйесін модульдер бойынша басқару; бағдарламаны тасымалдау әдісі бойынша, есептеу қалпынап күту қалпына, дайын күйге және керісінше, негізгі ресурстарды орналастыру(бөлу) әдісі бойынша: жадылар мен процессорлар). Басқа жүйелік бағдарламалық модульдер- транзиттік;

3) ОЖ- нің генерирлеу принципі. Есептеу комплексі мен шығарылатын есептер (LINUX) айналымының нақты конфигурациясынан шыға отырып, осы жүйенің супервизорлы бөлігін келтіруге болатын, ОЖ- нің центрлік жүйелік басқаруын анықтайтын тәсіл;

4) функционалдык артықшылық принципі. Бір сол жұмыстың әртүрлі әдістермен өткізілетін мүмкіндікті ескереді;

5) виртуализациялау принципі. Айырықша түгелденген немесе тұтас көрінісі- виртуалды машина;

6) сыртқы құрылыстан бағдарламалардың принципі;

7) бірлестік принципі- бұл басқа ОЖ үшін жазылған ОЖ-нің бағдарламаларды орындау мүмкіндігі. Бірлестік мынадай болады: бастанқы мәтін түрінде және екілік білестікте;

8) ОЖ-нің ашық және өсірмелі принципі. Қолданушылар мен жүйелік мамандарына да, анализ үшін ашық ОЖ ұйғарылады. Өсірмелі ОЖ генерацияның мүмкіндігін қолдануға ғана жасамай, сонымен қатар оның жаңа модульдердің құрамын енгізеді (клиент – сервер типі бойынша ОЖ құрамдауында негізделеді, қолайлылықпен қамтамасыз етеді). Ашық ОЖ-ға UNIX-жүйесі де жағады;

9) ауысу принципі – бұ бір процессор типінен екінші процессор типіне ауысуы. Бұл дегеніміз, ОЖ жоғары деңгейлі стандартты тілде жазылуы тиіс (Си тілінде);

10. қауіпсіздікті қамтамасыз ету принципі. Міндетті түрде қауіпсіздік ережелерді , қауіпсіздік облысының негізгі стандарттарын, әртүрлі деңгейдегі қауіпсіздікті анықтауды талап етеді.

Қазіргі заманның ОЖ-дің көпшілігі С2 деңгейлерге сәйкес келеді:

1) құпиялық кірудің амалдары, қолданушының идентификациясын қамтамасыз етеді. (ерекше есім мен парольді енгізу);

2) рұқсат етудің таңдамалы бақылауы;

3) есепке алу және бақылау құрылысы;

4) жадыны қорғау.

Микроядро – көбінесе басымдылық жағдайда (критикалық) жұмыс істейтін және сервистік қосымша ретінде қарастырылатын ОЖ қалған бөлігін қолдайтын ОЖ минималды стержендік бөлігі.

Микроядро тек қана 5 қызмет түрін қамтамасыз етеді:

- 1) виртуалды жадыны басқару;
- 2) есептер және ағымдар;
- 3) процессор аралық коммуникация;
- 4) шығару-енгізуді және үзілістерді қолдауды бақылау;
- 5) процессор мен хосттың қызметтер тізімі.

Host - IP – адресі бар басты компьютер. Монолитті ядро әрқайсысы бір-бірін шақыра алатын процедуралардан тұрады. Процедуралардың барлығы басымдылық режимінде жұмыс істейді. Сонымен монолитті ядро дегеніміз процедураларды тікелей шақыру арқылы өзара байланысатын және мәліметтердің жалпы құрылысын қолданатын, компоненттерінің барлығы бір бағдарламаның құрама бөліктері болып келетін операциялық жүйенің схемасы. Монолитті операциялық жүйе үшін ядро бүкіл жүйемен дәл келеді.

Монолитті ядросы бар операциялық жүйелердің көбінде ядроны жинау, яғни оның компиляциясы операциялық жүйе орнатылған әрбір компьютерге бөлек жасалады. Және де ядроға қосылатын құрылғылардың және бағдарламалық протоколдардың тізімін таңдауға болады. Ядро жалғыз бағдарлама болғандықтан оған жаңа элементтер қосу және қолданылмайтындарды алып тастау қайта компиляциялау көмегімен ғана іске асады. Ядро әрқашан оперативтік жадыда болғандықтан ядроға артық компоненттердің болмағаны дұрыс. Және де керек емес компоненттердің жойылуы операциялық жүйенің сенімділігін арттырады.

Монолитті ядро деп – операциялық жүйелерді ұйымдастырудың ескі тәсілі. Монолитті ядросы жүйелерге көбінесе Unix жүйелерін мысал ретінде келтіруге болады.

Монолитті (макроядролы) ОЖ – құрымдандырылған болғанымен жүйелік функциялады өзгерту және қосу кезінде қиыншылықтарға әкеп соқтырады.

Нақты уақыттағы ОЖ-ге қойылатын талаптар:

- 1) НУОЖ мультибағдарламалы және мультиесепті болуы керек;
- 2) ағымның басымдылығы туралы түсінік;
- 3) ОЖ-де приоритеттердің мұралану жүйесі болуы тиіс;
- 4) ОЖ есептің синхронизациялаудың мықты механизммен қамтамасыз етуі қажет;
- 5) ОЖ тәртібі белгілі және нақты болжаулы болуы тиіс.

Мысалдар: атомдық электростанцияларды, технологиялық процесстерді, ғарыштық навигацияларды басқару.

ОЖ интерфейсстермен құрастыру принциптері

ОЖ аппаратура мен тұтынушы арасындағы интерфейс ретінде

карастылады.

Есептің нақты сұрақтарына беретін ОЖ орындайтын функциялар:

1) процесстерді басқару:

а) іске қосу, тоқтату және есепті орындаудан алып тастау;

б) есеп басымдылығын өзгету немесе енгізу;

в) есептердің бір-бірімен байланысы (сигналдар механизмі, конвейрлер, пошта жәшіктері);

г) RPC (Remote Procedure Call) – бағдарламаны шақыру.

2) жадыны басқару:

а) жады блогын беру беруге сұраныс;

б) жадыны босату;

в) жады блогының параметрлерін өзгерту (мысалы, жады процесспен бекітіліп болуы мүмкін, немесе жалпы рұқсатта ұсынылады);

г) жады блогының параметрлерін өзгету (барлық жүйелерде бола бермейді).

3) Енгізу/ шығаруды басқару:

а) виртуалды құрылғыларды басқаруға сұраныс (енгізу/ шығару басқару ОЖ ерекшеленген функциясы болып табылады және тұтынушылық есептің ешқайсысы құрылғыларды тікелей басқаруға мүмкіндігі болмауы керек);

б) файлдық операциялар (файлда ұйымдастырылған мәліметтерді құру, өзгерту және өшіру үшін файлды басқару жүйесіне сұраныс жасау).

Командаларды сөйкес тілде қабылдап алып, оларды жүйелік интерфейстің негізгі шақыруларына түрлендіретін арнайы бағдарламалық модульдер іске асырылады. Әдетте бұл модульдерді команданың интерпритаторы деп атайды (мысалы, MS – DOS үшін – command.com, Windows 2000 үшін- cmd.exe). Тұтынушыдан команда алып, лексикалық және синтаксистік анализ жүргігеннен кейін, бұл модуль өзі немесе көп жағдайда ОЖ басқа модульдеріне API механизмін қолдана отырып байланысады. Сәйкесінше API механизмі бар ОЖ керекті параметрлерді беретін бағдарламаларды шақыру арқылы және де бағдарламалық үзулер механизм арқылы іске асырылуы мүмкін (DOS:API негізгі функциясы Int 21h кіру нүктесі арқылы ғана мүмкін).

API келесі бағыттарға бөлінеді:

1) жоғары деңгейдегі API интерфейсi, RTL (Runtime Library) кітапханасына қатысы бар- орындаудың уақыттық кітапханасы компиляциялау кезінде бағдарламалау жүйесі қолданатын қарапайым

бағдарламалардан тұрады;

- 2) ОЖ-ге кіретін қолданбалы және жүйелік API бағдарламасы;
- 3) басқа да API.

API іске асырудың бірнеше әдістері бар:

- 1) ОЖ деңгейіндегі іске асыру;
- 2) жүйелік бағдарламалау деңгейіндегі іске асыру;
- 3) процедура мен функцияның сыртқы кітапханасының деңгейіндегі іске асыру.

ОЖ деңгейінде API функциясын іске асырылуына ОЖ жауапты болады. Функцияларды орындайтын объектілік код ОЖ құрамына тікелей кіреді немесе берілген ОЖ арналып құрастырылған динамикалық жүктелгіш кітапхана құрамында болады. Жүйелік бағдарламалау осы кодты шақыратын интерфейс ұйымдастыру үшін ғана жауапты. Бұл жағдайда қорытындылы бағдарлама ОЖ тікелей байланысады, сондықтан API іске асырудың басқа варианттарына қарағанда API функциясын орындау көбіне эффективті болады.

Кемпілігі – қорытындылау бағдарламасы кодының ауыспалылығының болмауы ғана емес сонымен қоса бастапқы бағдарламаның кодының болмауы.

API тұтынушыға RTL бағдарламалау тіліне сәйкес функцияның кітапханасы ретінде беріледі. API өнімділігі біраз төменірек болады, API көптеген функцияларын орындау үшін RTL кітапханасы бәрібір ОЖ функцияларына сұраныс жасау керек. Бағдарламалаудағы бастапқы кодтың ауыспалығы бұл әдісте ең жоғарғы болады.

API функциясын сыртқы кітапхана көмегімен іске асыру. Сыртқы кітапхана ОЖ функцияларына да, RTL бағдарламалау тілінің функцияларына да сұрағыс жасағандықтан бұл әдіс көбіне өніміз болады.

Негізгі әдебиет: [1, б. 278-310]. Қосымша әдебиет: [9, б. 191-219].

Бақылау сұрақтары

1. ОЖ құрудың негізгі принциптерін атаңыз және оларды түсіндіріңіз?

2. ОЖ ядросы дегеніміз не?

3. ОЖ микроядросы қандай негізгі функция орындау керек?

4. API іске асырудың қандай түрлері бар?

5. ОЖ ақпараттық қауіпсіздігін қамтамасыз ету мәселесінде қойылатын негізгі талаптарды атаңыз.

Глоссарийлер

Буфер – ол 532 байт орын алатын, аралық көрсеткіштерді сақтауға қолданылатын оперативтік жадылық блок.

Буферизация – файл үлкен көлемді блоктармен оперативтік жадының ішкі буферіне көшіріледі, содан кейін көшірілген буфер дискідегі шығатын файлға жазылады.

Компьютордағы енгізу-шығару базалық жүйесі – BIOS (Basic Input Output System) құрамында базалық нұсқауға кіретін барлық құрылғыны қамтитын драйвер.

Үзіліс векторы – үзіліс өңдеушісінің адресін анықтауға қызмет ететін орын.

Көптапсырмалы ығысым – ОЖ орындалу құрылысының рұқсатынсыз процесстерге басқару жасай алады .

Виртуалдық адрес кеңістігі - виртуалдық адрес процесстерінің жиынтығы.

Есептік жүйе – тапсырманы орындау ақпаратын өңдейтін процесінің өзара әрекеті, ЭЕМ-дегі аппараттық жиынтығы мен бағдарламалық құралы.

Диспетчеризация – жоспарлау нәтижесінде тапқан тұжырымның шынайылығы.

Драйвер құрылғысы – жұмысты құрылғымен бірге талап ете алатын, пайдаланушы арқылы тапсырыстың алмасуын орындайтын, құрылғымен бірге термендейтін және құрылғыдан үзілісті өңдейтін статистикалық көрсеткіштегі көпкірісті бағдарламалық модуль.

Интерфейс – құрылғының өзара әрекеттік ережесінің жиынтығы және өзін немесе пайдаланушы арқылы өзара әрекетті іске асыратын бағдарлама.

Квант – ағынның орындалу жеткілігінде процесстік уақытының шектік үзіліссіз периоды.

Процесстің контексті – бөлу моментіндегі операциялық органың компьютер мен параметр аспабының бейнесіндегі аппараттық құрылымы.

Кластер – файлға айыратын жадының минималдық адрестік бірлігі (логикалық).

Монитор – көрсеткіштері және процедуралары бар керекті ортақ ресурсты динамикалық реттеудің қарама-қарсы шынайы механизмалық ұйымы.

Мьютекс – белгіленген немесе белгіленбеген екі күйдің біреуінде орналаса алатын жай ғана келген екілік семафор.

Үзіліс – бағдарламаның орындалу командасында ағымдағы

күйін, мәліметтерді сақтау арқылы, уақытша тоқтатылымы және арнайы меңгеру бағдарламаның үзілімін өңдеу.

Примитив – операциялық жүйенің базалық функциясы.

Процесс – пайдаланған көрсеткіштердің бөлшектерімен бірге бағдарламаның орындалу кезеңіндегі операцияның жалғасы.

Цул – реттеу объектілерінің динамикалық, біртекті жиынтығы, мысалы жады блогының ұзындығы бірдей болғанда.

Семафор – тек қана екі «ашылу» және «жабылу» операциялары үшін жүргізген параллельдік процесске өтімді ауыспалы арнайы тип.

Супернайдаланушы – өтімге кез-келген ресурсы шектелмеген. құқы бар пайдаланушы.

Синхронизация – осы оқиғаның басталу барысындағы жылдам жолмен ағынның тоқталуындағы келісімдегі тұжырымдан, кейбір оқиғаның басталуына және активизацияға дейінгі өзара әрекеттік механизмдік басқару.

Жүйелік шақырым – ОЖ ядросына қосымша сауал (запрос).

Операциялық жүйенің үйлестімі – ОЖ-нің қасиеттері басқа бір ОЖ-ге қосымша орындайтын мүмкіндіктерін қамтамасыз етеді.

Утилитга – бөлек тапсырмаларды шешетін басқармалар және есептік жүйеде бірге болуының бағдарламасы.

Файл – көрсеткіштерді есептейтін және жазып ала алатын сыртқы жадының аталатын облысы.

Эмулятор – арнайы бағдарламаны қамтамасыз ету, компьютерге машиналық нұсқауды интерпреттеуге рұқсат ететін, пайдаланушыдан жақсы нәтиже беретін, компьютер үшін жазылған архитектуралық процесорлар.

Зертханалық жұмыстар

№ 1 Microsoft Windows жүйесінің консольдық режиміндегі жұмысы

Тапсырма

1 Microsoft Windows операциялық жүйесінің негізгі командаларымен танысу.

2 Microsoft Windows жүйесінің консольдық режиміндегі жұмысын талдау. Филтрлерді, бағыт ауыстырушыларды және конверсерлерді қолдануды қамтитын өзіндік жұмыстарды жасау.

Негізгі әдебиет: [1, б. 362-363]

Қосымша әдебиет: [17, б. 38-57]

Бақылау сұрақтары

1. Microsoft Windows операциялық жүйесінің қандай негізгі командаларын білесіз?

2. Ішкі және сыртқы командалардың айырмашылығы неде?

3. Қандай жағдайда консольдық режимде жұмыс істеу қажеттілігі туатынын атап өтіңіз.

4. Филтрлер деі еніміз не?

5. Конверсер қашан қолданылады?

6. Бағытты ауыстыру көмегімен қандай есептерді шешуге болады?

№ 2 DISK EDITOR төмен деңгейлі редакторды талдау

Тапсырма

1 DISK EDITOR төмен деңгейлі редактордың әртүрлі режимдағы жұмысымен танысу.

2 Boot Record-тың құрамын көру.

3 C дискісінің түпкі каталогының құрамын көру. Берілген тапсырма бойынша берілген файлдың орын алатын кластер санын есептеу.

4 FAT тізбегін қарау, есептелген мәнді көргенмен салыстыру.

5 «Физикалық сектор» режимінде C дискісінің жүктелу секторын қарау: 16-лық кодтауда мұғалімнің сұрағына жауап беру.

Негізгі әдебиет: [11, б. 63-208], [4, б. 56-67]
Қосымша әдебиет: [9, б. 146-191], [13, б. 224-262]

Бакылау сұрактары

1. Сектор көлемі қай жерде бейнеленеді?
2. View ішкі менюдің командаларын сипаттаңыз
3. Link ішкі менюдің командалары не үшін қызмет етеді?
4. Диск құрамын қалай түзетуге болады?

№ 3 FAT12. FAT32 файлдық жүйелерін талдау

Тапсырма

1 DE көмегімен FAT12. FAT32-ге арналған резервтік аумақтың құрамын талдау, қорытынды жасау.

2 Command.com және Bootlog файлдарының FAT тізбегін салыстыру, қорытынды жасау.

3 Талдаған файлдарды del командасының көмегімен жою, каталогта, FAT тізбегінде және кластерларда болған өзгерістерді сипаттау. Сөйкес команданың көмегімен файлдарды қайта орнына келтіру.

Негізгі әдебиет: [11, б. 63-208], [4, б. 56-67]
Қосымша әдебиет: [9, б. 146-191], [13, б. 224-262]

Бакылау сұрактары

1 FAT12. FAT32 файлдық жүйелерінің негізгі айырмашылықтары қандай?

2 Кластер түсінігіне анықтама беріңіз.

3 Файлды дискке жазуды бастайтын бірінші кластердің номері туралы ақпарат қайда жазылады?

4 FAT жүйесіндегі әрбір кластердің жағдай қай жерде анықталады?

5 FSInfo не үшін қызмет етеді?

№ 4 NTFS файлдық жүйесін талдау

1 DISK PROBE төмен деңгейлі редактордың жұмысымен

танысу

2 NTFS жүктелгіш секторының құрамымен танысу (DISK PROBE көмегімен жүктелгіш сектордың дамын алу), бөлімнің геометриясын анықтау.

3 MFT басты таблицаны іздеуді екі әдіспен іске асыру.

4 MFT таблицасындағы жазулар құрамымен танысу.

5 DISKPROBE колдана отырып өз варианттарына сәйкес екі тапсырманы орындау. 1-вариант үшін тапсырмаларға мысал:

1) тапсырма. Ерекше есіммен текстік файлға ярлық құраңыз.

2) тапсырма. Ерекше есіммен файл құрыңыз және оған 14 белгімен 50жол енгізіңіз.

3) тапсырма. Бірінші және екінші тапсырмаларға MFT таблицасының дами секторының жазуын қарастырыңыз.

Негізгі әдебиет : [1, б. 163-208], [4, б. 56-67], [6, б. 224-262].

Қосымша әдебиет : [9, б. 146-191], [13, б. 224-262].

Бақылау сұрақтары

1 NTFS енгізу секторының бөлімінде бастапқы облыс нені құрайды?

2. Қандай жағдайларда \$MFT файлында кеңейтілген жазу болады?

3. Төмендегі аталғандарда қандай ақпарат болады:

а) стандарттық атрибутте;

б) тізім атрибуттарында;

в) мәліметтер атрибуттарында?

4. Резиденттік атрибуттың резиденттіктен айырмашылығы қандай. Қандай атрибуттар үнемі резидентті?

№ 5 Қатты дисктің логикалық структурасы

Тапсырма

1 DISKEDITOR көмегімен қатты дисктің бірінші секторын опалғылық түрде көріп, ақпараттың активтілік бөлімін табу, бастапқы және соңғы бөлімдердің адрестары, бөлімнің типі.

2 D логикалық дисктің таблицасын қарастыру: дисктің күнделікті сектордың басталуы туралы ақпаратты табу, D дискінің.

3 FDISK эмулятордың конфигурирлеу бағдарламасының көмегімен қатты диск бойынша оздік жұмысты орындау

Тапсырмаға мысал: бастапқы қалпы: Бір ҚД, өлшемі 20 Гб.
Негізгі бөлімнің өлшемі – 5 Гб, кеңейтілгені - 10 Гб.

Қабылдайтын қалпы: Негізгі бөлім - 10 Гб, қосымша - 10 Гб. Үш
ЛД-тің өлшемдері 3,3 и 4 Гб. Бірінші бөлім активті болып табылады.

Негізгі әдебиет: [1, б. 145-156]

Қосымша әдебиет: [9, б. 147-156], [14, б. 82-151], [15, б. 383-
387]

Бақылау сұрақтары

1. Қатты дисктің бірінші бөлімінде не орналасқан?
2. Басты жазу жүктелуінің негізгі функциялары қандай?
3. Partition Table-дің құрамына не жатады ?
4. Бөлім таблицаларында өдетте бөлімдер қандай тәртіппен сипатталады (Partition Table-да)?
5. Қатты дисктің бірінші секторыдағы сигнатураның маңызы қандай?
6. Сипаттау бөлімінде қандай ақпарат орналасқан?
7. Бірінші бөлімнің сипаттаушысы қай адреске сілтеп тұр?
8. Кеңейтілген бөлімнің сипаттаушысы қай бөлімге сілтеп тұр?
9. SMBR-дің бірінші сипаттаушысы қай адреске сілтеп тұр ?
10. SMBR –дің екінші сипаттаушысы қай адреске сілтеп ?
11. Физикалық және логикалық секторлардың айырмашылығын аңқтаңыз?

№ 6 Командалық файлдар

Тапсырма

Командалық файлдардың негізгі командаларымен танысу.

Командалық файлдарды құру бойынша өздік жұмысты орындау.

Негізгі әдебиет: [1, б. 145-156]

Қосымша әдебиет: [8, б. 300-309], [17, б. 549-553]

Бақылау сұрақтары

1. ОЖ-нің пакеттік файлында хабарлар қандай жағдайларда іске асады?
2. Пакеттік файлдарда тоқталым қандай жағдайларда іске асады?

3. Пакеттік файлдарда шарттың анализі қандай жағдайларда іске асады?

4. Пакеттік файлдарда диалог қалай ұйымдастырылады?

5. Пакеттік файлдарда шартты анализ қалай ұйымдастырылады?

№ 7 Реестрді оқын зерттеу

Тапсырма

1 regedit көмегімен реестрдің структурасын оқып тану.

2 Реестрдің параметрлерін өзгерте отырып, өз вариантың бойынша оздік жұмысты орындау.

Тапсырмаға мысал:

Жолдық параметр құра отырып, оның көмегімен ОЖ Windows жүктелгеннен кейін автоматты түрде Microsoft Word жүктеледі. Тапсырманы орындап болғаннан кейін Microsoft Word жүктеу параметрін автоматты түрде өшіру.

Негізгі әдебиет: [1, б. 145-156]

Қосымша әдебиет: [12, б. 136-145].

Бақылау сұрақтары

1. Windows 98 реестрдің структурасы қандай?

2. Тармақ дегеніміз не, бөлім және ішкі бөлім, реестрдің құрамына неше тармақ кіреді ?

3. Параметр дегеніміз не, олар қандай бөлімдерден тұрады?

4. Реестрдің резервирлеу және қалыныптастыру әдістерінің кез келгені мазмұндан беріңіз .

5. Реестрден басқа, жүйенің жүктелу кезінде автоматты түрде жүктелетін бағдарламалар туралы мағлұматтар қайда сақталады?

6. ПУСК батырмасы менюің стандартты пункттерін өшіру және қайта қалыптастыруы қалай іске асырылады ?

7. Windows қабықшасы мен жұмыс үстелінің әртүрлі орнатуларды келтіру қай ішкі бөлімде орналасқан?

Өзіңді-өзің тексеру үшін арналған тесттік тапсырмалар

1 Операциялық жүйе дегеніміз не?

- A) ДК-нің түйіндерінің белгілі бір жиынтығы.
- B) Жаңа бағдарламаларды жасауға мүмкіндік беретін бағдарламалар комплекті.
- C) ДК жұмысын жүргізетін бағдарламалар комплексі.
- D) Пайдаланушының жұмысын жақсартуға арналған бағдарлама-қабықшасы.
- E) Қазіргі оқиғаға кепілдік қайтару уақытын қамтамасыз ететін ОЖ.

2 Операциялық жүйенің түсінігі.

- A) Операция жиынтығы.
- B) Техникалық жүйе.
- C) Еңгізу-шығару жүйесі.
- D) Операцияны іске асыру үшін арналған аспап аппарат.
- E) Бағдарламаның комплексі .

3 MS DOS-та ағымды каталогтың мазмұнын экранға шығаратын бұйрық:

- A) DIR.
- B) REN.
- C) CD.
- D) RD.
- E) MD.

4 MS DOS ОЖ-де ағымды каталогта CATALOG каталогын жасайтын бұйрық:

- A) DEL CATALOG.
- B) MD CATALOG.
- C) RD CATALOG.
- D) TYPE CATALOG.
- E) DIR CATALOG.

5 Windows 95, 98, XP жатады:

- A) Драйверлерге.
- B) Операциялық жүйелерге.
- C) Қолданбалы бағдарламаларға.
- D) Операциялық қабықшаларға.
- E) Утилиттер.

6 Драйверлер – бұл:

- A) Дискте кателерді іздейтін бағдарламалар.
- B) Белгілі бір есепті шығаруға бейімделген бағдарламалар.
- C) Сыртқы құрылғылармен жұмыс істеуге бейімделген арнайы түрдегі бағдарламалары.
- D) Стандартты интерфейс.
- E) Кіру-шығудың негіздік жүйесі.

7 Қай драйвердің көмегімен клавиатурадан мәліметтер енгізіледі және олардың дисплей экранына шығарылуы жүргізіледі?

- A) CONFIG.SYS.
- B) IO.SYS.
- C) MSDOS.SYS.
- D) ANSISYS.
- E) IO.EXE.

8 Жүйенің жіберілуі жазбасынан кейін жадқа қандай жүйелік файлдар енгізіледі?

- A) IO.SYS, MSDOS.SYS И COMMAND.COM.
- B) CONFIG.SYS, MSDOS.SYS И COMMAND.COM.
- C) IO.SYS, ANSISYS, CONFIG.SYS.
- D) IO.SYS, ANSISYS, COMMAND.COM.
- E) IO.EXE, ANSISYS, CONFIG.SYS.

9 MS DOS ОЖ-де тамыр каталогына қайту үшін қандай бұйрықты орындау қажет:

- A) RDL.
- B) CD.
- C) DIR/P.
- D) MD/P.
- E) TYPE/P.

10 MS DOS үшін дискетаның бір секторында қанша байт бар:

- A) 1024 байт.
- B) 2048 байт.
- C) 512 байт.
- D) 248 байт.
- E) 128 байт.

11 BIOS қайда ориаласады?

- A) Аналық платасында.
- B) Операциялық жүйесінің еңгізгішінде.
- C) Слоттарда.
- D) Жалпы бағдарламаларда.
- E) Винчестерде.

12 BIOS құрамындағы қандай арнайы бағдарламасы PC жағу кезінде орындалады?

- A) ADInf.
- B) Aidstest.
- C) POST.
- D) FileFind.
- E) ADExc.

13 Ядро модулі

- A) Тұрақты жедел жадыда болады.
- B) Өз жұмысын орындау барысында ғана жедел жадыға тиеледі.
- C) Регистрлік жадыда болады.
- D) Сыртқы жадыда болады.
- E) Слоттарда.

14 AUTOEXEC.BAT – бұл

- A) ПК құрылғыларын тестілеу үшін орындалатын файл.
- B) ОЖ-ні жүктегенде автоматты түрде жүктелетін командалық файл.
- C) Сыртқы құрылғыларға рұқсатты қадағалайтын файл
- D) Файлдарды берілген тәртіппен сорттаудың жолын жазатын файл.
- E) Архивация тәртібі жазылатын файл.

15 CONFIG.SYS-бұл

- A) Берілген ПК-ның аппараттық құралдарын ың жұмыс тәртібін беретін командалар құрамында болатын файл.
- B) Файлдарды берілген тәртіппен сорттаудың жолын жазатын файл.
- C) Архивация тәртібі жазылатын файл.
- D) Сыртқы құрылғыларға рұқсатты қадағалайтын файл.
- E) ОС-ті жүктегенде автоматты түрде жүктелетін командалық файл.

16 Компьютер жұмыстары принциптерін алғашқы рет кім қисынға келтірді?

- A) Паскаль.
- B) Нейман.
- C) Лейбниц.
- D) Бэббидж.
- E) Гольдстейн.

17 Процессордың ырақты жиілігі немен өлшенеді?

- A) об/с.
- B) Мгц.
- C) бит/сек.
- D) байт/сек.
- E) Кбайт.

18 486 DX4*100/RAM 8 M/HDD 520 M/FDD 3.5"/SVGA 1M кескін үйлесімімен компьютер процессорының ырақты жиілігі қандай болады?

- A) 100.
- B) 8.
- C) 12.
- D) 520.
- E) 82.

19 486 DX4*100/RAM 8 M/HDD 520 M/FDD 3.5"/SVGA 1M кескін үйлесімімен компьютер процессорының оперативті жадысы қандай болады?

- A) 8.
- B) 4.
- C) 1.
- D) 2.
- E) 5.

20 486 DX4*100/RAM 8 M/HDD 520 M/FDD 3.5"/SVGA 1M кескін үйлесімімен компьютер процессорының қатты дискінің көлемі қандай болады?

- A) 100.
- B) 486.
- C) 520.
- D) 440.
- E) 256.

21 486 DX4*100/RAM 8 M/HDD 520 M/FDD 3.5"/SVGA 1M кескін үйлесімімен компьютер процессорының видео жадының көлемі қандай болады?

- A) 1Mб.
- B) 3.5 Mб.
- C) 8 Mб.
- D) 4.8 Mб.
- E) 3 Mб.

22 486 DX4*100/RAM 8 M/HDD 520 M/FDD 3.5"/SVGA 1M кескін үйлесімімен компьютер процессорының видеоадаптердің түрі қандай болады?

- A) EGA.
- B) SVGA.
- C) VGA.
- D) SEGA.
- E) SMGA.

23 Орныдалатын файл қандай кеңейту бола алады ма?

- A) .ARJ, .DAT.
- B) .COM, .EXE.
- C) .DOC, .TXT.
- D) .PAS, .XLS.
- E) .SYS, .MDB.

24 Уақытты талдау ОЖ дегеніміз не?

- A) ЭЕМ-ге бірнеше пайдаланушыны қамтамасыз ететін диалог өткізгіш бөліміндегі ОЖ.
- B) ЭЕМ-ге уақыт интервалымен қиылыспайтын, бірнеше пайдаланушыны қамтамасыз ететін өткізгіштегі ОЖ.
- C) ЭЕМ-ге уақыт моменті қиылыспайтын реттік диалог өткізгішін қамтамасыз ететін ОЖ.
- D) ЭЕМ-ге кезекті өткізгішті қамтамасыз ететін ОЖ.
- E) ЭЕМ-ге бірнеше пайдаланушыны бір уақытта қамтамасыз ететін диалог өткізгішіндегі ОЖ.

25 Шынайы уақыттағы ОЖ дегеніміз не?

- A) Қазіргі оқиғаға кепілдік қайтару уақытын қамтамасыз ететін ОЖ.
- B) Белгілі бір момент уақытында процес өткізгішін пландамайтын ОЖ.

- С) FIFO стратегиялық күтуші ретінде қолданылатын ОЖ.
- Д) LRU стратегиялық күтуші ретінде қолданылатын ОЖ.
- Е) Қазіргі оқиғаға кепілдік қайтару уақытын қамтамасыз етпейтін ОЖ.

26 Баспаңқы Bootstrap жүктеу бағдарламасы қай жерде орналасқан.

- А) Directory' да.
- В) Көрсеткіштер мекен-жайы.
- С) Логикалық дисктің алғашқы секторында (сектор №0).
- Д) FAT'те.
- Е) Boot Record'да.

27 (Boot Record) дискісі бойынша параметрлік кесте қай жерде орналасқан?

- А) Directory'да.
 - В) Көрсеткіштер мекен-жайы.
 - С) Логикалық дисктің алғашқы секторында (сектор №0).
 - Д) FAT'те.
 - Е) FS Info'да.
- Жауабы: С)

28 BPB (FAT12, FAT32) параметрі қай жерде орналасқан?

- А) Directory'да.
- В) Көрсеткіштер мекен-жайында.
- С) Логикалық дисктің алғашқы мекен-жайында (сектор №0).
- Д) FAT'те.
- Е) FS Info'да.

29 (FAT32) алғашқы бос кластер номері туралы мәліметтер қай жерде орналасқан?

- А) Directory'да.
- В) Көрсеткіштер мекен-жайы.
- С) Логикалық дисктің алғашқы мекен-жайында (сектор №0).
- Д) FAT'те.
- Е) FS Info'да.

30 (FAT32) дискісіндегі бос кластерлерінің мөлшері туралы мәлімет қайда орналасқан?

- А) Directory'да.
- В) Көрсеткіштер мекен-жайы.

- C) Логикалық дисктің алғашқы мекен-жайында (сектор №0).
- D) FAT'те.
- E) FS Info'да.

31 (FAT32) қалыптасу уақытында қандай шектес облыстарында дисктер құртылады?

- A) Резервтік облыста, FAT, Directory, көрсеткіштер мекен-жайы.
- B) Резервтік облыста, FAT.
- C) Bootstrap, FAT, Directory.
- D) Резервтік облыста, FAT, көрсеткіштер мекен-жайы.
- E) Резервтік облыста, FAT, Directory.

32 FAT32 файлдық жүйе құру уақытында қандай жүйелік облыстар жасалынады?

- A) Резервтік облыста, FAT, Directory, көрсеткіштер мекен-жайы.
- B) Резервтік облыста, FAT.
- C) Bootstrap, FAT, Directory.
- D) Резервтік облыста, FAT, көрсеткіштер мекен-жайы.
- E) Резервтік облыста, FAT, Directory.

33 (FAT12, FAT32 файлдық жүйе) маңызды файлының көрсеткіштері қай жерде азылған?

- A) Резервтік мекен-жайы.
- B) Boot Record'да.
- C) Каталог ағымында.
- D) Көрсеткіштер мекен-жайы.
- E) FATте.

34 (FAT16, FAT32 файлдық жүйе) файлы дискке жазылғандығы туралы мәлімет қай жерде бекітіледі?

- A) Boot Record, FAT.
- B) Резервтік облыста, FAT.
- C) Резервтік мекен-жайы, Directory.
- D) Каталог ағымында, FAT.
- E) Каталог ағымында, резервтік мекен-жайы.

35 Көрсеткіштер облысында кластердің максималдық саны немен шектес ?

- A) FAT бөсендігімен.
- B) Тамыр каталогының өлшемімен.
- C) Сектор өлшемімен.

- D) Жолдың ұзындығымен.
- E) FAT көлемімен.

36 Қатты дисктің бірінші секторында не орналасқан?

- A) Boot Area, Partition Table, Сигнатура.
- B) (MSB) жүктеу жазылымымен, (Partition table) кесте бөлімімен, жүктеу секторының типтік сигнатурасы
- C) (MSB) жүктеу жазылымымен, типтік сигнатура.
- D) Boot Record, Boot Area, Boot Strap.
- E) MSB, Boot Strap, сигнатура.

37 (MSB) бастапқы жүктеу жазылымының негізгі функциясы не?

- A) Primary бөліміне басқарма жібереді.
- B) Кеңейтілген бөлімнің басына басқарма жібереді.
- C) Partition Table-ді сканерлеп, активтік бөлігін анықтайды және соның басына басқарманы жібереді.
- D) Жүйелік файлдарды іздейді.
- E) Кесте бөлігінің сигнатурасын талдайды.

38 Partition Table дегеніміз не?

- A) Екі бөліктің суреттемесі.
- B) Операциялық жүйенің дискке жазылғандығы туралы мәлімет.
- C) Төрт бөліктің суреттемесі.
- D) Барлық бөліктің суреттемесі.
- E) Логикалық диск туралы мәлімет.

39 Windows 98 файл атының ең үлкен ұзындығы?

- A) 255 символ.
- B) 8 символ.
- C) 12 символ.
- D) 128 символ.
- E) Windows 98 файл атының ең үлкен ұзындығына шектеуліктер жоқ.

40 SMBR-да екінші жазылымы қай мекен-жайы сілтейді.

- A) FSINFO мекен-жайы.
- B) Boot Record мекен-жайы.
- C) Логикалық дискінің бастапқысындағы мекен-жайы.
- D) SMBR-дегі келесі логикалық дискінің мекен-жайы.

Е) Bootstrap мекен-жайы.

41 SMBR-дің орналасуы туралы мәлімет қай жерде көрсетілген?

- А) логикалық секторларда.
- В) физикалық секторларда.
- С) кластерлерде.
- Д) байттарда.
- Е) параграфтарда.

42 Операциялық аймақ дегеніміз не?

- А) Негізгі жат.
- В) ОЖ-ге қоңіл бөлетін керекті бағдарламаның және пайдаланушының интерфейсі.
- С) Еңгізу-шығару аспабы.
- Д) Сырттай есте сақтайтын құрылғы.
- Е) Жүйелік орта.

43 Ағын дегеніміз не?

- А) Ағын ішкі параллелизм процесін қолдануды қажет етеді.
- В) Ағын процесстер аралығындағы байланысты құрайды.
- С) Ағын процессаралығындағы бөліктің алмасуын құрайды.
- Д) Ағын бір бағдарламаның ішіне бірнеше операцияның түрін орындауға мүмкіндік береді.
- Е) Ағын процессаралығындағы ақпаратты ауыстыруға мүмкіндік береді.

44 Процесс дегеніміз не?

- А) Буфердегілерді дискіге көшіру.
- В) ОЗУ тесттері.
- С) Бағдарламаның орындалу кезеңі.
- Д) Бір күйден екінші күйге көшірілімі.
- Е) ОЖ-нің ВЗУ-ға қатынасы.

45 Кооперативтік көптапсырмалар дегеніміз не?

- А) Барлық тапсырмалар, бірдей алмасу буферінің пайдалануын айтады.
- В) Әрбір тапсырма МП-дегі қолдану уақытын өзі анықтайтынын айтады.
- С) ОЖ әр тапсырмада МП-дегі қолдану уақытын анықтайтынын айтады.

D) Барлық тапсырмалар бір уақыттағы кеңістікте орындалуын айтады.

E) Барлық тапсырмаларға бірдей, жүктелу файлдарының айырылымын айтады.

46 Ыгілсімды контапсырмалар дегеніміз не?

A) Барлық тапсырмалар, бірдей алмасу буферінің пайдалануын айтады.

B) Әрбір тапсырма МП-дегі қолдану уақытын өзі анықтайтынын айтады.

C) ОЖ әр тапсырмада МП-дегі қолдану уақытын анықтайтынын айтады.

D) Барлық тапсырмалар бір уақыттағы кеңістікте орындалуын айтады.

E) Барлық тапсырмаларға бірдей, жүктелу файлдарының айырылымын айтады.

47 Оверлей дегеніміз не?

A) Сегменттер қолданылатын құрылым жады.

B) ОП тек бір уақытта бағдарламаның басты бөлігі және бір немесе бірнеше жаппайтын сегменттерде орналаса алатын құрылым жады.

C) Бағдарламаның магниттік дискке уақытша, түгелдей тиелетін құрылымдық жады.

D) Бағдарламаның электрондық дискіге жартылай тиелетін құрылым жады.

E) Беттерді пайдаланатын құрылым жады.

48 Свонпінг дегеніміз не?

A) Сегменттер қолданылатын құрылым жады.

B) ОП тек бір уақытта бағдарламаның басты бөлігі және бір немесе бірнеше жаппайтын сегменттерде орналаса алатын құрылым жады.

C) Бағдарламаның магниттік дискке уақытша, түгелдей тиелетін құрылымдық жады.

D) Бағдарламаның электрондық дискіге жартылай тиелетін құрылым жады.

E) Беттерді пайдаланатын құрылым жады.

49 Физикалық жады дегеніміз не?

A) Сегменттің реттік жиынтығы.

- В) Кластердің реттік жиынтығы.
- С) Беттің реттік жиынтығы.
- Д) Көптеген ұяшықтың реттері (ішкелген).
- Е) Бит жиынтығы.

50 Виртуалдық адрес дегеніміз не?

А) Виртуалдық дискідегі командалардың және ауысымдардың адрестері.

В) Виртуалдық машинадағы командалардың және ауысымдардың адрестері.

С) Оверлейлік бағдарламадағы командалардың және ауысымдардың адрестері.

Д) Жүйелік бағдарламаның орындалу дайындығы мен дайын машиналық бағдарламадағы командалардың және ауысымдардың адрестері.

Е) Бағдарламаның объектік модуліндегі командалардың және ауысымдардың адрестері.

«Операциялық жүйелер» пәні бойынша дұрыс жауабы көрсетілген

№	Жауабы	№	Жауабы
1	С	26	С
2	Е	27	С
3	А	28	С
4	В	29	Е
5	В	30	Е
6	С	31	Д
7	Д	32	В
8	А	33	Д
9	В	34	Д
10	С	35	А
11	А	36	В
12	С	37	С
13	А	38	С
14	В	39	А
15	А	40	Д
16	В	41	В
17	В	42	В
18	А	43	А
19	А	44	С
20	С	45	В
21	А	46	С
22	В	47	В
23	В	48	С
24	Е	49	Д
25	А	50	Д

Әдебиеттер

Негізгі

- 1 Гордеев А. В. Операционные системы. – 2-е издание. СПб. : Питер, 2004. – 416 с.
- 2 Дейтел Г. Введение в операционные системы. – М. : Мир, 1987. – 358 с.
- 3 Вильям Столлингс. Операционные системы. Четвертое издание. – М. : Издательский дом «Вильямс», 2002. – 452 с.
- 4 Карпов В. Е., Коньков К. А. Основы операционных систем М. : изд-во ИНТУИТ, 2004. – 200 с.
- 5 Таненбаум Э. Современные операционные системы. СПб.: Издательский дом Питер, 2002. – 459 с.
- 6 Робачевский А. Операционная система UNIX Спб.: BHV, 1999. – 641 с.

Қосымша

- 7 Богумирский Б.С. Руководство пользователя ПЭВМ. 4 1,2. – Санк-Петербург: Печатный двор, 1994. – 352 с.
- 8 Гордеев А.В., Молчанов А.Ю. Системное программное обеспечение.- СПб: Питер, 2001. – 736 с.
- 9 Данкан Р. Профессиональная работа в MS – DOS': Пер. с англ.- М.: Мир, 1993.- 509 с., ил.1992. – 416 с.
- 10 Мюллер С. Модернизация и ремонт персональных компьютеров. Пер. с англ.- М.: Восточная Книжная Компания, 1996. – 396 с.
- 11 Д. Бэкон, Харрис. Операционные системы. Издательская группа BHV 2004. – 825 с.
- 12 Alex Workman. Новый способ установки 20 операционных систем на одном компьютере. М. : 2005. – 368 с.
- 13 Кулаков В. Программирование на аппаратном уровне. Специальный справочник. – СПб: Питер, 2001.- 496 с.
- 14 Шалин П. Реестр Windows XP. Спец. справочник. Санк-Петербург, Питер 2003.
- 15 Чеботарев И., Климов А. Реестр Windows.<http://www.whatis.ru> 2004
- 16 http://www.carnie.angara.ru/recovery/8_ntfs_data.htm

Мазмұны

Кіріспе	3
1 Операциялық жүйелердің мәні мен функциялары	4
1.1 Операциялық жүйелердің классификациясы	4
2 Файлды жүйелер	8
2.1 Негізгі анықтамалар. Ұйымдастыру принциптері	8
2.2 Файлдар жүйесі. FAT типті файлдар жүйесі бар логикалық диск (том)	13
2.3 NTFS типті файл жүйесінің логикалық дискі (том)	17
3 Қатты дискінің логикалық құрылымы	22
3.1 Қатты дискідегі деректердің ұйымдастырылуы	22
3.2 Операциялық жүйедегі енгізу-шығаруды басқару	25
4 Операциялық орта	30
4.1 Есептеу процесі және процестерде жүргізілетін операциялар ресурсы	30
5 Жадымен басқару	37
5.1 Жады және бейнелер, виртуалды адрестік кеңістік	37
5.2 Жадымен басқару. Сегменттік, беттік және сегментті-беттік жадыны ұйымдастыру	39
5.3 Жадымен басқару. Жаңа компьютерлерде жадының көптүрлілігі	42
6 ОЖ архитектурасы және қолданбалы бағдарламалаудың интерфейсі	45
Глоссарийлер	50
Зертханалық жұмыстар	52
Өзінді-өзін тексеру үшін арналған тесттік тапсырмалар	57
Әдебиеттер тізімі	68

БЕКІТЕМІН

С. Торайғыров атындағы
Павлодар мемлекеттік
университетінің ректоры
жөніндегі проректор




Құрастырған: аға оқытушы Исабеков Жанат Бейсембаевич

Есептеу техникасы және бағдарламау кафедрасы

Операциялық жүйлер

050704 – Есептеу техникасы және бағдарламалық қамтама білім беру
мамандықтары студенттеріне арналған оқу-әдістемелік құралы

Кафедра мәжілісінде бекітілді 2009 ж. «22» желтоқсан № 4 хаттама

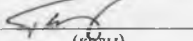
Кафедра меңгерушісі  О. Г. Потапенко
(колы)

“Физика, математика және ақпараттық технологиялар” факультетінің оқу
әдістемелік кеңесінде мақұлданды

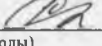
2010 ж. « 26 » 01 № 6 хаттама

ОӘК төрайымы  А. Т. Кишубаева
(колы)

КЕЛІСІЛДІ


ФМ және ЛТ факультет деканы  С. К. Тлеуменов
(колы)

2010 ж. « 6 » 01

СМ бөлімі п/б  Г. С. Баяхметова
(колы)

2010 ж. « 31 » 03

МАҚҰЛДАНДЫ

ОҰЖ ж ӘҚБ бастығы  А.А. Варакута
(колы)

2010 ж « 01 » 04

Ж.Б. Исабеков

ОПЕРЦИЯЛЫҚ ЖҮЙЕЛЕР

Оқу - әдістемелік құралы

Техникалық редактор Г.Н. Сейтахметова
Жауапты хатшы Г.К. Туkenова

Басуға 20.04.2010 ж.
Әріп түрі Times.
Пішім 29,7 x 42 ¼. Офсеттік қағаз.
Шартты баспа табағы 2,71 Таралымы 20 дана
Тапсырыс № 1241

«КЕРЕКУ» Баспасы
С.Торайғыров атындағы
Павлодар мемлекеттік университеті
140008, Павлодар қ., Ломов к., 64