Дәріс 9. Виртуалды жад

9.1. Аппараттық және басқару құрылымдары

Жергілікті және виртуалды жад
Бетті ұйымдастыру
Сегменттеу
Сегменттеу және бетті ұйымдастыру комбинациясы

9.2. Операциялық жүйенің бағдарламалық жасақтамасы

Іріктеу стратегиясы
Орналастыру стратегиясы
Ауыстыру стратегиясы
Резиденттік жиынтығын басқару
Тазалау стратегиясы
Жүктеуді басқару

9.1. Аппараттық және басқару құрылымдары

Бетті ұйымдастыру және сегменттеу кезінде жадты басқару келесі сипаттамаларға ие.

1. Процесс шеңберіндегі барлық жад сілтемелері-бұл жұмыс уақытында физикалық мекен-жайларға динамикалық түрде таратылатын логикалық мекен-жайлар. Бұл процесті дискіге түсіріп, негізгі жадқа қайта жүктеуге болатындығын білдіреді, сондықтан жұмыс уақытының әртүрлі нүктелерінде ол негізгі жадтың әртүрлі жерлерінде болуы мүмкін.
2. Процесті негізгі жадта бірыңғай үздіксіз блокпен орналастыруға міндетті емес бірқатар бөліктерге (беттерге немесе сегменттерге) бөлуге болады. Бұл динамикалық мекен-жайларды аудару және беттер немесе сегменттер кестесін пайдалану арқылы қамтамасыз етіледі.

Сондықтан процестің барлық беттерінің немесе сегменттерінің негізгі жадта бір уақытта болуы міндетті шарт емес. Егер келесі таңдалған команда орналасқан фрагмент (сегмент немесе бет) және бағдарлама кіретін жад ұяшығы орналасқан фрагмент- негізгі жадта болса, онда бағдарламаны орындау кем дегенде біраз уақытқа созылуы мүмкін.

Жаңа процесті жадқа жүктеу уақыты келді делік. Амалдық жүйе оны жадқа тек бір немесе бірнеше блоктарды (беттер немесе сегменттер), соның ішінде бағдарламаның басталуы бар блокты орналастырудан бастайды. Белгілі бір уақытта негізгі жадта орналасқан процестің бөлігі процестің резиденттік жиынтығы (resident set) деп аталады. Процесс барысында барлық сілтемелер тек процестің резиденттік жиынтығына қатысты болған сияқты болады. Сегменттер немесе беттер кестесін қолдана отырып, процессор әрқашан өңдеуді қажет ететін блоктың негізгі жадта орналасқанын анықтай алады. Егер процессор негізгі жадта жоқ логикалық мекен-жайға тап болса, ол жадқа кіру

қатесін көрсететін үзіліс жасайды. Амалдық жүйе бұзылған процесті бұғатталған күйге аударады және басқаруды алады. Тоқтатылған процесті жалғастыру үшін амалдық жүйе негізгі жадқа логикалық мекен-жайы бар блокты жүктеуі керек. Ол үшін амалдық жүйе дискіден оқу сұрауын қолданады (жұмыс уақытында басқа процестер жалғасуы мүмкін). Қажетті блок негізгі жадқа жүктелгеннен кейін, басқаруды операциялық жүйеге беретін кіріс-шығыс үзілісі жасалады, ол өз кезегінде бұғатталған процесті дайындық күйіне ауыстырады.

Жаңа стратегияны қолданудың екі салдары бар.

1. Негізгі жадта көптеген процестерді қолдауға болады. Негізгі жадқа әр процестің кейбір блоктары жүктелгендіктен, жадқа көбірек процестерді орналастыруға болады. Бұл өз кезегінде процессорды тиімді пайдалануға әкеледі, өйткені кез-келген уақытта белсенді процестердің болу ықтималдығы артады.
2. Процесс барлық негізгі жадтан үлкен болуы мүмкін. Бағдарламалаудағы ең маңызды шектеулердің бірі жеңілді. Бағдарлама (және бағдарламашы) сонымен бірге үлкен жад көлеміне қол жетімді, бұл диск кеңістігінің мөлшері. Қажет болса, Операциялық жүйе қажетті процесс блоктарын жадқа автоматты түрде жүктейді.

Процесс тек негізгі жадта орындалатындықтан, бұл жад нақты жад деп те аталады (Нақты жад). Алайда, бағдарламашы немесе қолданушы дискіде бөлінген әлдеқайда үлкен жадпен айналысады. Бұл жад виртуалды (виртуалды жад) ретінде белгілі. Виртуалды жад-Бұл жадты бөлу схемасы, онда жадты негізгі жадтың бөлігі ретінде шешуге болады. Виртуалды жад өте тиімді мультипликацияны қамтамасыз етеді және негізгі жад көлеміне қатысты қатаң шектеулерді алып тастау арқылы пайдаланушының жұмысын жеңілдетеді. Кестеде. 9.1 виртуалды жадты қолдана отырып және онсыз бетті ұйымдастырудың және сегментацияның негізгі сипаттамалары келтірілген.

Кесте 9.1. Бетті ұйымдастыру және сегменттеу сипаттамалары

Қарапайым бетті ұйымдастыру	Виртуалды жады бар бетті ұйымдастыру	Қарапайым сегментация	Виртуалды жадпен Сегментация
Негізгі жад кадрлар деп аталатын бекітілген өлшемдегі шағын блоктарға бөлінеді. Бағдарлама компилятор немесе жадты басқару жүйесі арқылы парақтарға бөлінген. Кадрлардағы ішкі фрагментация.		Негізгі жад бөлінбейді. Бағдарлама сегменттерін компиляция кезінде бағдарламашы анықтайды (сегменттерге бөлу туралы шешімді бағдарламашы қабылдайды). Ішкі фрагментация жоқ.	

<table>
<tr><td colspan="2">Сыртқы фрагментация жоқ.

Амалдық жүйе әр процесс үшін бет кестесін қолдауы керек, бұл процестің осы бетінде қай кадр бос емес екенін көрсетеді.

Операциялық жүйе бос кадрлардың тізімін қолдауы керек.

Абсолютті мекенжайды есептеу үшін процессор бет нөмірі мен офсетті пайдаланады.</td><td colspan="2">Сыртқы фрагментация.

Амалдық жүйе жүктеу мекен-жайы мен әр сегменттің ұзындығын көрсететін әр процесс үшін сегменттер кестесін қолдауы керек.

Амалдық жүйе бос жад блоктарының тізімін қолдауы керек.

Абсолютті мекенжайды есептеу үшін процессор сегмент нөмірін және офсетті пайдаланады.</td></tr>
<tr><td>Процесс жұмыс істеуі үшін оның барлық парақтары негізгі жадта болуы керек (қабаттасуды қолданудан басқа).</td><td>Процесс жұмыс істеуі үшін оның барлық парақтары негізгі жадта болмауы керек; қажет болған жағдайда оларды жүктеуге болады.

Бетті негізгі жадқа оқу бетті дискіге жазуды қажет етуі мүмкін</td><td>Процесс үшін оның барлық сегменттері негізгі жадта болуы керек (қабаттасуды қолданудан басқа)</td><td>Процесс жұмыс істеуі үшін оның барлық сегменттері негізгі жадта болмауы керек; қажет болған жағдайда оларды жүктеуге болады.

Негізгі жадқа сегментті оқу дискіге бір немесе бірнеше сегменттерді жазуды қажет етуі мүмкін.</td></tr>
</table>

Жергілікті және виртуалды жад

Осылайша, уақыттың әр сәтінде негізгі жадта белгілі бір процестің аз ғана бөлігі болады, сондықтан негізгі жадта бір уақытта көптеген процестер орналастырылуы мүмкін. Кейбір блокты негізгі жадқа жүктеген кезде, басқа блокты сол жерден түсіру керек. Егер сіз бірден қажет болатын блокты жадтан шығарсаңыз, амалдық жүйе сол блоктарды негізгі жадқа және дискіге тұрақты жылжытумен айналысады. Мұндай аударымдардың көп болуы өткізу қабілетінің төмендеуі (thrashing) деп аталатын жағдайға әкеледі: процессор негізінен процестерді орындаумен емес, негізгі жадқа түсіру және жүктеумен айналысады. Бұл жағымсыз әсерді жою міндеті 1970 жылдары орындалған және әртүрлі күрделі, бірақ тиімді алгоритмдердің пайда болуына әкелген бірқатар зерттеу жұмыстарына арналған. Шын мәнінде, олар жүйеде соңғы оқиғалар негізінде жақын болашақта қандай жад блоктары қажет екенін анықтауға тырысады. Бұл әдістер локализация принципіне негізделген [60],

онда код пен мәліметтерге жүгіну процесте кластерлеуге бейім екендігі айтылады. Әдетте, бағдарламаларда қайталанатын циклдар мен кіші бағдарламалар бар. Цикл немесе кіші кезек пайда болғаннан кейін, процессор тек бірнеше қайталанатын командалар жиынтығына жүгінеді. Кестелер мен массивтермен жұмыс топтастырылған деректерге қол жеткізуді де қамтиды. Уақыт өте келе кейбір қолданылатын кластерлер басқаларымен ауыстырылады, бірақ қысқа уақыт ішінде процессор негізінен тұрақты жад кластерлерімен жұмыс істейді. Осылайша, біраз уақыт жұмыс істеу үшін процестің аз ғана бөлігі қажет болады; сонымен қатар, жақын арада жұмыс істеу үшін процестің қандай бөліктері қажет болатындығы туралы дұрыс болжам жасауға болады, осылайша өткізу қабілетінің төмендеуіне жол берілмейді. Локализация принципі виртуалды жад схемасының тиімділігіне үміттенуге мүмкіндік береді. Бұл үшін, біріншіден, бетті ұйымдастыруға және/немесе сегментацияға аппараттық қолдау қажет, екіншіден, операциялық жүйеде екінші және негізгі жад арасындағы беттерді және/немесе сегменттерді жылжытуға арналған бағдарламалық жасақтама болуы керек.

Бетті ұйымдастыру

Қарапайым бетті ұйымдастыруды қарастырған кезде, әр процестің жеке парақ кестесі бар екендігі айтылды, ол процестің барлық беттерін негізгі жадқа жүктеген кезде жасалады. Беттер кестесіндегі әр жазбада жадтағы тиісті беттің кадр нөмірі болады. Процестердің әрқайсысына байланысты бірдей Парақ кестесі парақты ұйымдастыру негізінде виртуалды жадты ұйымдастыру кезінде қажет-алайда, бұл жағдайда кесте жазбаларының құрылымы біршама күрделі болады (сурет. 9.1, а). Негізгі жадта тек процестің кейбір парақтары болуы мүмкін болғандықтан, кестенің әр жазбасында негізгі жадта тиісті беттің болуын көрсететін р биті болуы керек. Егер бұл бет негізгі жадта болса, онда кесте жазбасында оның кадр нөмірі болады.

Виртуалды мекенжай

Бет нөмірі	Орын ауыстыру

Беттер кестесінің жазылуы

Р	М	Басқа басқару биттері	Кадр нөмірі

а)Беттік ұйымдастыру

Виртуалды мекенжай

Сегмент нөмірі	Орын ауыстыру

Сегменттер кестесінің жазылуы

Р	М	Басқа басқару биттері	Ұзындығы	Сегменттің бастапқы мекенжайы

б) Сегменттеу

Виртуалды мекенжай

Сегмент нөмірі	Бет нөмірі	Орын ауыстыру

Сегменттер кестесінің жазылуы

Басқару биттері	Ұзындығы	Сегменттің бастапқы мекенжайы

Беттер кестесінің жазылуы

Р	М	Басқа басқару биттері	Кадр нөмірі

Р-қатысу биті

М-бит модификациясы

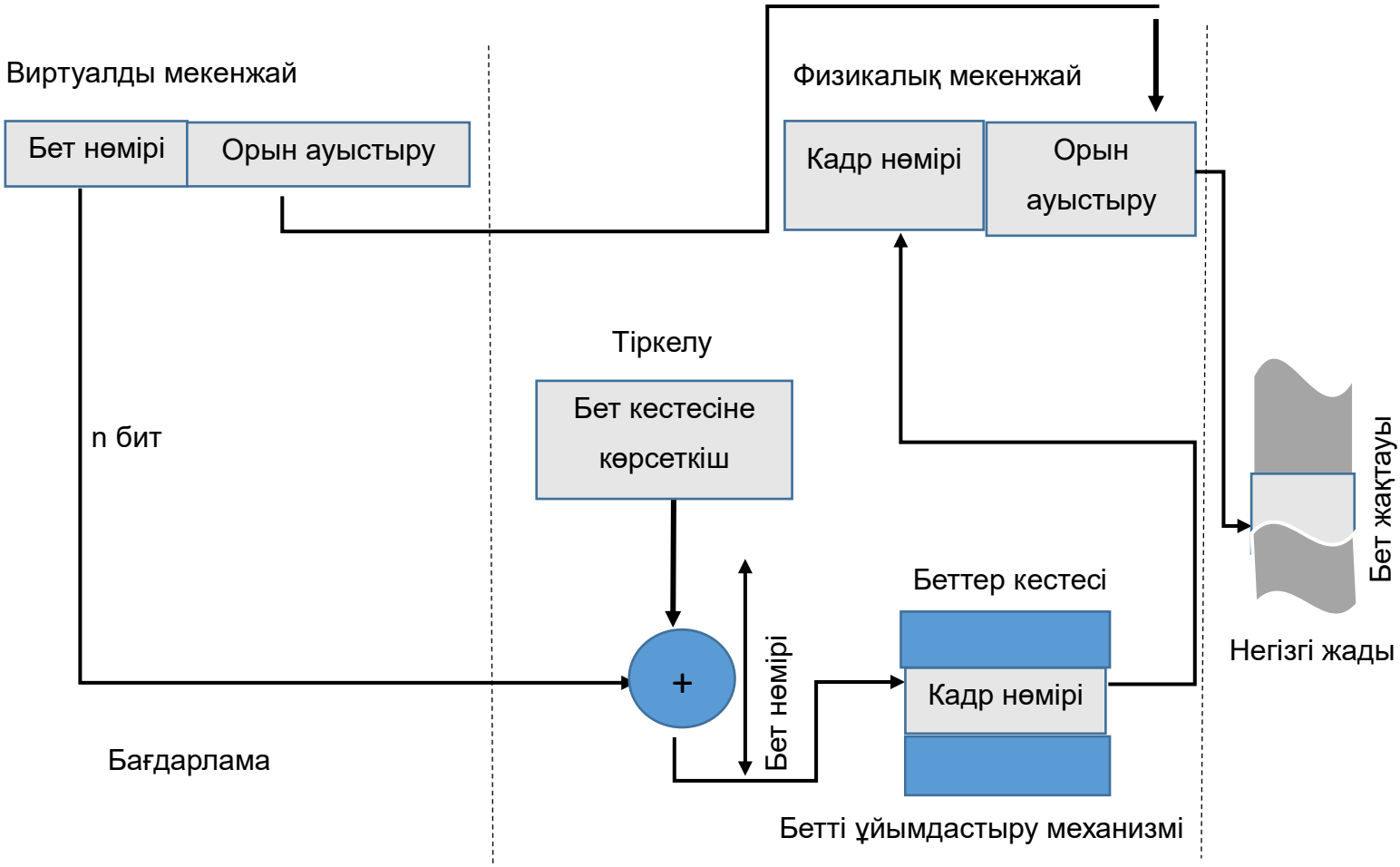
в) Бетті ұйымдастыру мен сегменттеудің үйлесімі

Сурет. 9.1. Жадты басқару жүйесінің типтік форматтары

Бет кестесінің жазбасындағы тағы бір басқару биті-модификация биті, М, ол негізгі жадқа соңғы жүктелгеннен бастап берілген беттің мазмұны өзгергенін көрсетеді. Егер ешқандай өзгерістер болмаса, онда қазіргі уақытта оның кадрындағы бетті ауыстыру уақыты келгенде, бұл бетті дискіге жазудың қажеті жоқ, өйткені дискіде оның дәл көшірмесі бар. Парақ кестесінің жазбасында басқа басқару биттері болуы мүмкін, мысалы, Парақ деңгейінде жадты қорғау немесе бөлісу мақсатында қызмет етеді.

Бет кестесінің құрылымы

Жадтан сөзді оқудың негізгі механизмі бет нөмірі мен офсеттен тұратын виртуалды немесе логикалық мекенжайды бет кестесін қолдана отырып, кадр нөмірі мен офсеттік болып табылатын физикалық мекен-жайға таратуды қамтиды. Бет кестесі процестің көлеміне байланысты өзгермелі ұзындыққа ие болғандықтан, оны регистрлерге орналастыру мүмкін емес және бет кестесі негізгі жадта орналасуы керек. Суретте. 9.2 бұл механизмнің аппараттық орындалуы көрсетілген. Кейбір процесті орындау кезінде оның бет кестесінің бастапқы мекен-жайы регистрде сақталады, ал виртуалды мекен-жайдағы бет нөмірі тиісті кадр нөмірін іздейтін элемент индексі ретінде қолданылады. Содан кейін бұл нөмір бізді қызықтыратын жад ұяшығының нақты физикалық мекенжайын алу үшін виртуалды мекен-жайдан ығысумен біріктіріледі.



Сурет. 9.2. Жүйеде мекен-жайды беттік ұйыммен тарату

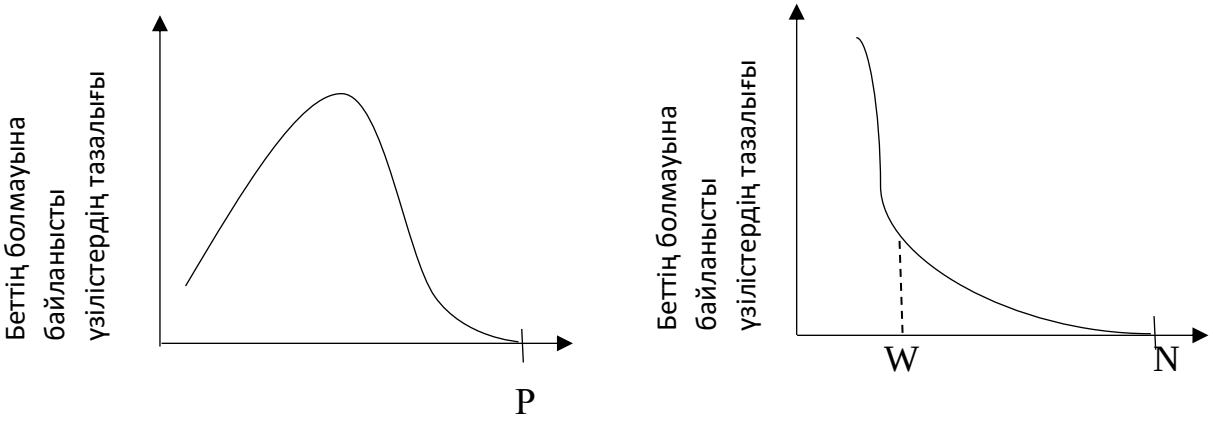
Көптеген жүйелерде әр процесс үшін бір парақ кестесі бар. Алайда, әр процесс виртуалды жадтың үлкен көлемін қолдана алады, сәйкесінше үлкен көлемдегі Беттер кестесі болады. Сондықтан виртуалды жад схемаларының көпшілігі Парақ кестелерін нақты емес, виртуалды жадта сақтайды. Бұл дегеніміз, Парақ кестелерінің өзі кез-келген басқа беттер сияқты Парақ ұйымының объектісіне айналады. Процесс жұмыс істеген кезде, оның Парақ кестесінің кем дегенде бір бөлігі негізгі жадта орналасуы керек, оның ішінде қазіргі уақытта орындалатын Парақ туралы жазба болуы керек.

Бет өлшемі

Даму кезінде өте маңызды мәселе-парақтың көлемін таңдау. Мұнда бірден бірнеше факторларды ескеру қажет. Олардың бірі-ішкі фрагментация. Негізгі жадты пайдалануды оңтайландыру үшін азайтылған ішкі фрагментация беттің көлеміне тікелей байланысты екені түсінікті.

Екінші жағынан, беттердің өлшемі неғұрлым аз болса, соғұрлым көп процесс қажет болады, бұл бет кестесін ұлғайтуды білдіреді. Жүктелген көп функциялы ортадағы үлкен бағдарламалар үшін бұл белсенді процестердің бет кестелерінің бір бөлігі виртуалды жадта болады және жадқа кірген кезде парақтың болмауына байланысты екі рет үзіліс болады: алдымен-Парақ кестесінен қажетті жазбаны алған кезде, содан кейін процесс бетіне жүгінген кезде. Тағы бір фактор-бұл көптеген екінші жад құрылғыларының физикалық сипаттамалары, нәтижесінде үлкен блоктардың берілуі тиімдірек болады.

Сұрақ сонымен қатар негізгі жадта парақтың болмауына байланысты үзілістің пайда болу жиілігіне беттің өлшемі әсер ететіндігімен күрделене түседі. Суретте. 9.3, және жергілікті принципін ескере отырып, парақтың болмауына байланысты үзілістердің әдеттегі жиілігі көрсетілген. Егер парақтың өлшемі өте кішкентай болса, онда процесс беттерінің салыстырмалы түрде көп саны жадқа орналастырылады.



а) Бет өлшемі б) Бет кадр жиілігі

P- процесс өлшемі
W- жұмыс жиынтығының өлшемі
N-процесс беттерінің жалпы саны

Сурет. 9.3. Беттік ұйымның типтік мінез-құлқы

Біраз уақыттан кейін жадтағы парақтарда процестің соңғы қол жетімділігіне жақын орналасқан бөліктері болады, ал парақтың болмауына байланысты үзілістің жиілігі аз болуы керек. Беттің өлшемі ұлғайған сайын, әр жеке парақта соңғы орындалған қоңыраулардан жадқа дейін орналасқан мәліметтер болады. Тиісінше, локальдық принцип әлсірейді және парақтың болмауына байланысты үзілістер санының өсуі байқалады. Ақыр соңында, парақтың өлшемі процестің өлшемімен салыстырыла бастағанда (диаграммадағы P нүктесі), парақтың болмауына байланысты үзілістер сирек кездеседі және осы процестің көлеміне жеткенде олар мүлдем тоқтайды.

Сондай-ақ, процеске бөлінген кадрлар санының әсерін ескеру қажет. Суретте. 9.3, б парақтың белгіленген өлшемі үшін парақтың болмауына байланысты үзілістердің жиілігі негізгі жадтағы беттер санының өсуімен төмендегенін көрсетеді. Осылайша, бағдарламалық жасақтама стратегиясына (процесске бөлінген жад көлемі) аппараттық шешім (бет өлшемі) әсер етеді.

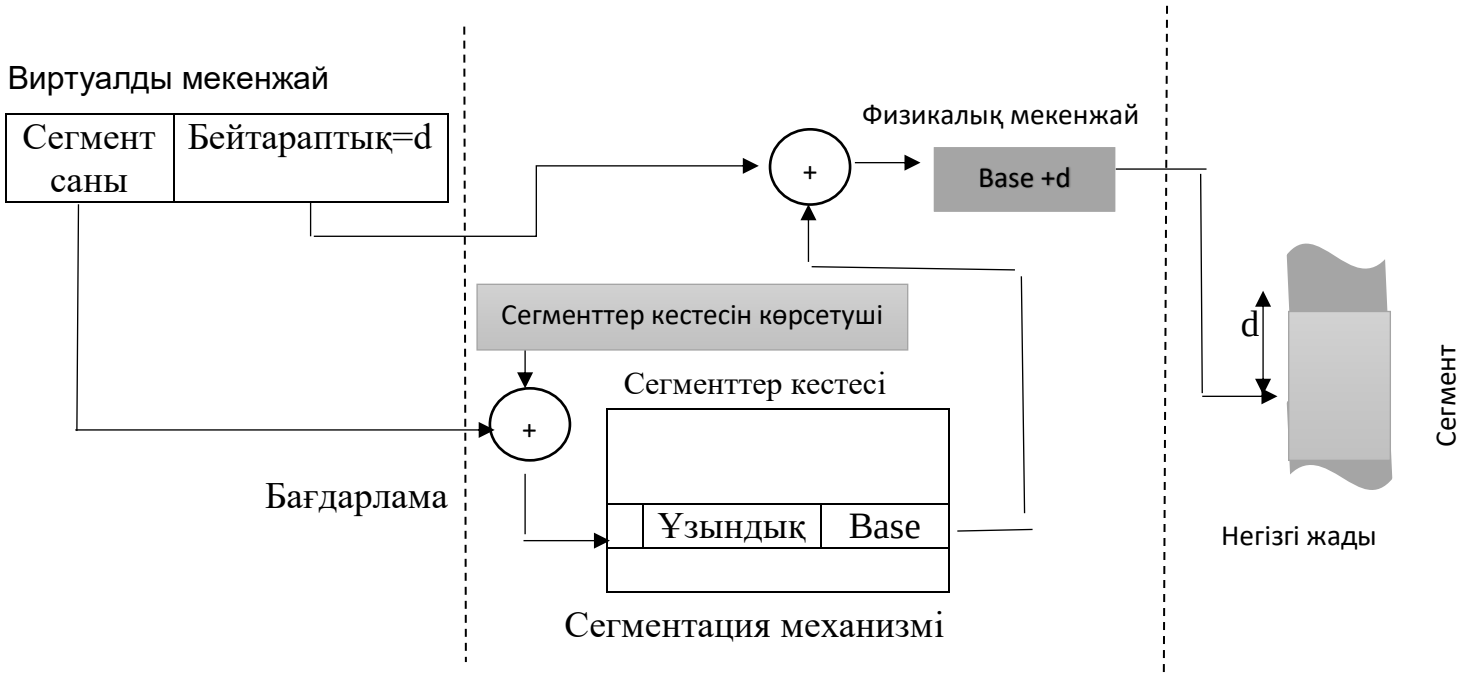
Кестеде. 9.2 кейбір машиналарда беттердің өлшемдері көрсетілген.

Кесте 9.2. Бет өлшемдерінің мысалдары

Компьютер	Бет саны
Atlas	512 48-биттік сөз
Honeywell-Multics	1024 36-биттік сөз
IBM 370/XA және 370/ESA	4 Кбайт
VAX Отбасы	512 байт
IBM AS/400	512 байт
DEC Alpha	8 Кбайт
MIPS	4 Кбайттан 16 Мбайтқа дейін
UltraSPARC	8 Кбайттан 4 Мбайтқа дейін
Pentium	4 Кбайттан 4 Мбайтқа дейін
Intel Itanium	4 Кбайттан 256 Мбайтқа дейін
Intel core i7	4 Кбайттан 1 Гбайтқа дейін

Сегменттеу

Сегменттеуге негізделген виртуалды жад схемасындағы әдеттегі әдіс-бұл әр процесс үшін жеке сегменттер кестесін қолдану (сурет. 9.1, 6). Жадтан сөзді оқудың негізгі механизмі сегменттер кестесін қолдана отырып, сегмент нөмірі мен ығысудан тұратын виртуалды немесе логикалық адресті физикалық мекен-жайға таратуды қамтиды. Сегменттер кестесінде процестің көлеміне байланысты өзгермелі ұзындық болғандықтан, сегменттер кестесін сақтау үшін негізгі жад қолданылады. Суретте. 9.4 сипатталған схеманы аппараттық іске асыру ұсынылады.



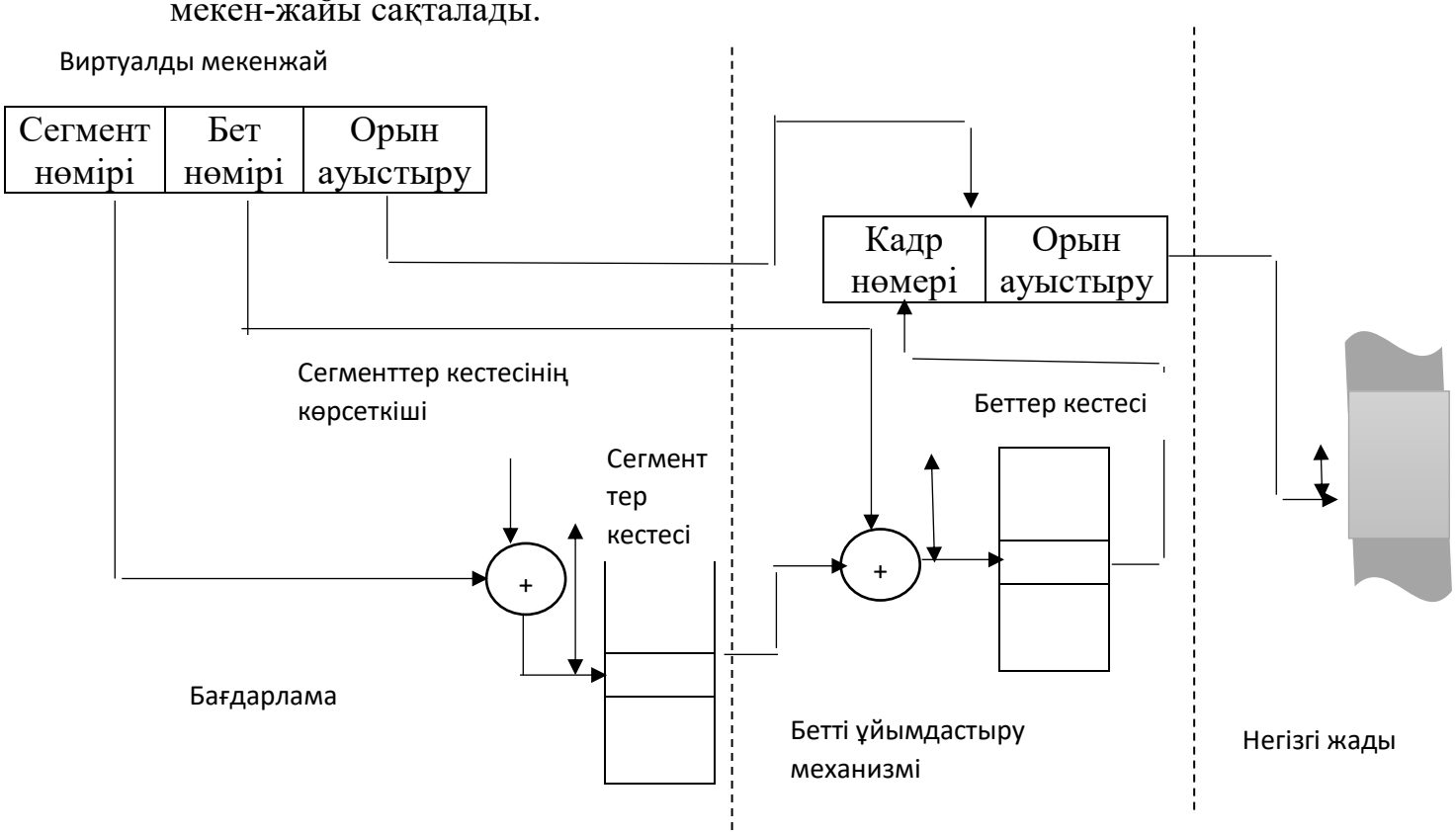
Сурет. 9.4. Сегменттеу жүйесінде мекенжайды аудару

Сегменттеу және бетті ұйымдастыру комбинациясы

Бетті ұйымдастырудың да, сегментацияның да артықшылықтары бар. Бағдарламашы үшін мөлдір бетті ұйымдастыру сыртқы фрагментацияны жояды және осылайша негізгі жадтың тиімді пайдаланылуын қамтамасыз етеді. Сонымен қатар, негізгі жадқа және одан жылжытылатын блоктар тұрақты, бірдей мөлшерде болғандықтан, жадты басқарудың тиімді алгоритмдерін құру оңайырақ болады. Сегменттеу бағдарламашы үшін көрінетін бола отырып, алдыңғы бөлімде көрсетілген артықшылықтарға ие, оның ішінде модульділік, өсіп келе жатқан деректер құрылымын өңдеу мүмкіндігі, сондай-ақ жадты бөлісу және қорғауды қолдау. Кейбір есептеу жүйелері тиісті аппараттық құралдармен және операциялық жүйемен жабдықталған, екі әдістің де артықшылықтарын пайдаланады.

Мұндай біріктірілген жүйеде пайдаланушының мекен-жай кеңістігі бағдарламашының қалауы бойынша бірқатар сегменттерге бөлінеді. Әр сегмент, өз кезегінде, негізгі жад жақтауының өлшеміне сәйкес келетін белгіленген өлшемдегі бірқатар беттерге бөлінеді. Егер сегменттің өлшемі беттің өлшемінен аз болса, ол бүкіл бетті алады. Бағдарламашының көзқарасы бойынша, бұл жағдайда логикалық мекен-жай сегмент нөмірінен және ондағы орын ауыстырудан тұрады. Операциялық жүйенің позициясынан сегменттегі орын ауыстыру белгілі бір сегменттің бет нөмірі және ондағы орын ауыстыру ретінде қарастырылуы керек.

Суретте. 9.5 сегменттеу мен бетті ұйымдастырудың үйлесімін қолдау үшін құрылым ұсынылады. Әр процесспен бір сегмент кестесі және бірнеше (бір сегментке бір) парақ кестелері байланысты. Белгілі бір процесс жұмыс істеген кезде процессор регистрінде тиісті сегменттер кестесінің бастапқы мекен-жайы сақталады.



Сурет. 9.5. Сегменттеу мен бетті ұйымдастыруды бөлісу кезінде мекенжайды тарату

9.2. Операциялық жүйенің бағдарламалық жасақтамасы

Іріктеу стратегиясы

Іріктеу стратегиясы бетті негізгі жадқа қашан беру керектігін анықтайды. Екі негізгі нұсқа – сұраныс бойынша және алдын-ала. Сұраныс бойынша іріктеу кезінде бет осы бетте орналасқан жад ұяшығына сілтеме жасалған кезде ғана негізгі жадқа жіберіледі. Егер жадты басқару жүйесінің барлық басқа элементтері жақсы жұмыс істесе, онда келесі жағдайлар орын алуы керек. Процесс жаңа басталған кезде параққа қоңыраулардың үзілуі пайда болады, бірақ содан кейін локальдылық принципі іске қосылады және жақында жүктелген беттерге қоңыраулардың көбеюі орын алады. Тиісінше, парақтың болмауына байланысты үзілістер саны өте төмен деңгейге дейін азаяды.

Алдын ала іріктеме жағдайында айналымның үзілуін тудырған бет қана жүктелмейді. Алдын-ала іріктеу көптеген екінші жад құрылғыларының сипаттамаларын пайдаланады, мысалы, іздеу уақыты және Дискінің айналуымен байланысты кідірісі бар дискілер. Егер процесс парақтары екінші реттік жадта орналасса, онда сол беттерді бір уақытта бір-бірлеп жүктегеннен гөрі бірнеше дәйекті беттерді бір уақытта негізгі жадқа жүктеу әлдеқайда тиімді болады.

Орналастыру стратегиясы

Орналастыру стратегиясы процестің бөліктері физикалық жадта қай жерде орналасатынын анықтайды. "Таза" сегментация жағдайында орналастыру стратегиясы өте маңызды мәселе болып табылады, оның шешімдері бірінші қолайлы, келесі қолайлы және басқа стратегия түрінде болуы мүмкін. Алайда, тек бетті ұйымдастыруды немесе бетті ұйымдастыруды сегментациямен бірге қолданатын жүйелер үшін орналастыру стратегиясы әдетте соншалықты маңызды емес, өйткені мекен-жайдың аппараттық таратылуы және жадқа аппараттық қол жетімділік кез-келген "бет-кадр"комбинациясында бірдей тиімді.

Ауыстыру стратегиясы

Ауыстыру стратегиясы ауыстыру үшін осы жиынтықтан қандай беттерді таңдау керектігін шешу болып табылады. Негізгі жадтың барлық кадрлары бос емес және парақтың болмауына байланысты үзілісті өңдеу процесінде жаңа бетті орналастыру қажет болған кезде, ауыстыру стратегиясы жүктелген беттің жақтауын босату үшін қазіргі уақытта негізгі жадтағы қандай беттерді түсіру керектігін анықтайды. Барлық стратегиялар жақын арада жүгінбейтін бетті жүктеуге бағытталған. Локальдық принципке сәйкес, соңғы уақытта айналымдар болған көптеген беттер мен жақын арада айналымдар болатын көптеген беттер арасында тығыз байланыс бар. Осылайша, көптеген стратегиялар бағдарламаның болашақ мінез-құлқын оның өткен мінез-құлқына сүйене отырып анықтауға тырысады. Әр түрлі стратегияларды

қарастыру кезінде стратегия неғұрлым жетілдірілген және ақылды алгоритм қолданса, оны жүзеге асыру кезінде үстеме шығындар соғұрлым жоғары болатындығын ескеру қажет.

Кадрларды бұғаттау

Әр түрлі алгоритмдерді қарастырмас бұрын, бір шектеуді атап өту керек: негізгі жадтың кейбір кадрлары бұғатталуы мүмкін. Жақтауды құлыптау қазіргі уақытта осы кадрда сақталған бетті ауыстыру мүмкін емес дегенді білдіреді. Операциялық жүйелердің көптеген ядролары негізгі басқару құрылымдары сияқты Құлыпталған кадрларда сақталады. Сонымен қатар, негізгі жадтың Құлыпталған жақтауларында кіріс-шығыс буферлері және қол жеткізу уақытына қатысты басқа да маңызды деректер мен код орналасуы мүмкін. Бұғаттау әр кадрға тиісті битті орнату арқылы жүзеге асырылады. Бұл бит кадрлар кестесінде де болуы мүмкін және қазіргі бет кестесіне қосылуы мүмкін.

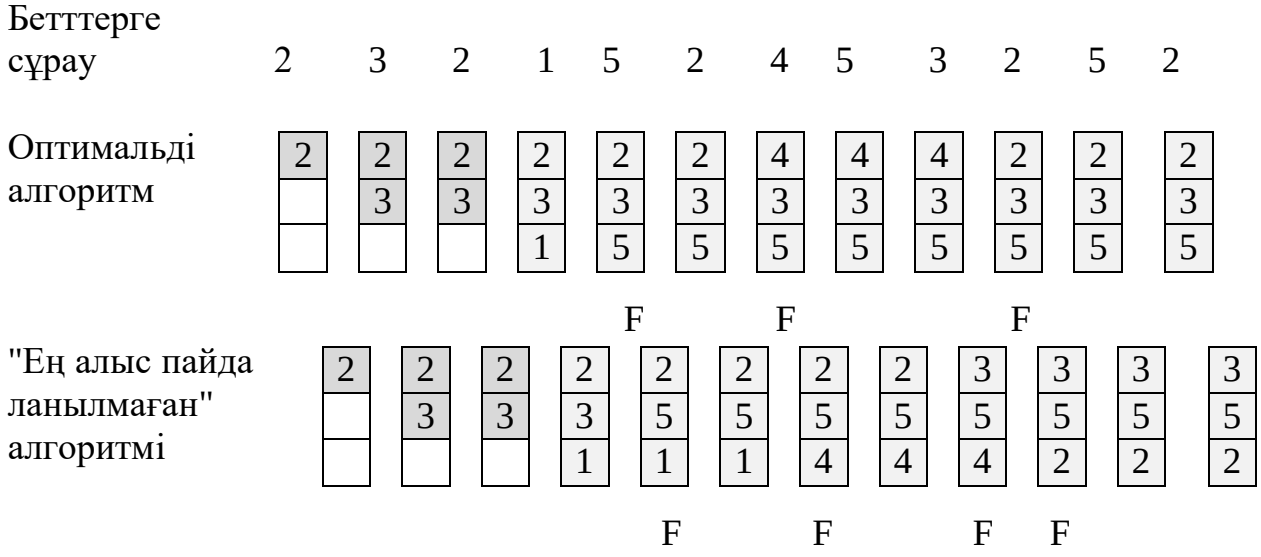
Негізгі Алгоритмдер

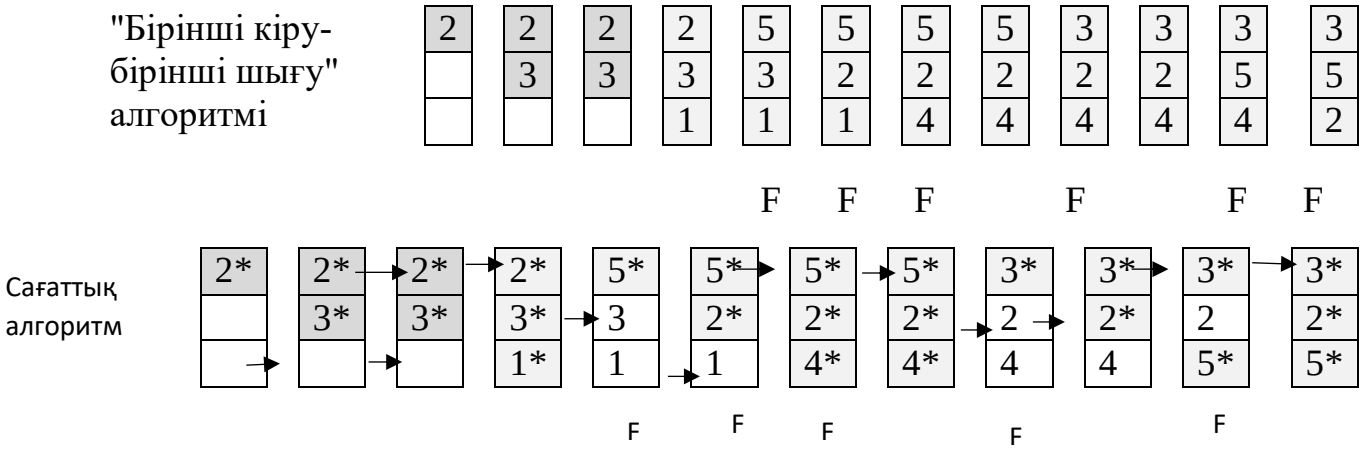
Ауыстырылатын бетті таңдау үшін қолданылатын бірқатар негізгі алгоритмдер бар.

- Оңтайлы алгоритм.
- Алгоритм барлық пайдаланылмаған элементтен ұзағырақ
- Алгоритм "бірінші кірді - бірінші шықты".
- Сағат алгоритмі

Оңтайлы стратегия-бұл барлық басқа парақтармен салыстырғанда ең көп уақыт аралығында болатын бетті ауыстыру үшін таңдау. Бұл алгоритм беттің болмауына байланысты ең аз үзілістерге әкелетінін көрсетуге болады [20]. Мұндай алгоритмді жүзеге асыру мүмкін емес екені түсінікті, өйткені бұл жүйе Болашақ барлық оқиғаларды білуі керек. Алайда, бұл алгоритм нақты алгоритмдерді салыстыратын стандарт болып табылады.

Суретте. 9.6 оңтайлы стратегияның мысалы келтірілген.





F-кадрларды бастапқы толтырғаннан кейін бетке жүгінуді үзу

Сурет 9.6. Бетті ауыстырудың төрт алгоритмінің әрекеті

Бұл процесс үшін кадрларды белгіленген бөлу (үш кадрдан тұратын резиденттік жиынның белгіленген мөлшері) пайдаланылады деп болжануда. Процесті орындау бес түрлі параққа жүгінуге әкеледі. Жұмыс барысында беттерге жүгіну келесі тәртіпте орындалады:

2 3 2 1 5 2 4 5 3 2 5 2

Бұл дегеніміз, алдымен 2 – бетке, содан кейін 3-бетке және т.б. оңтайлы стратегия кадрлар жиынтығын толтырғаннан кейін параққа үш рет үзілуге әкеледі (суретте "F"әріптерімен көрсетілген).

Стратегия пайдаланылмаған элементтің бәрінен ұзақ, жадта басқаларға қарағанда ұзақ қоңыраулары жоқ бетті ауыстырады. Жергілікті принципке сәйкес, бұл бет жақын арада пайдаланылмайды деп күтуге болады. Бұл стратегия шын мәнінде оңтайлы емес. Негізгі мәселе-оны жүзеге асырудың күрделілігі. Іске асыру нұсқаларының бірі параққа соңғы жүгіну уақытын белгілеуді қамтиды; бұл әр жадқа жүгінген кезде жасалуы керек, қандай өтініш орындалғанына қарамастан – кодқа немесе деректерге. Бұл схеманы аппараттық қолдау жағдайында да үстеме шығындар тым үлкен. Тағы бір нұсқа парақтарға қоңырау шалу бумасын сақтауды қамтиды, бұл жүйенің өнімділігі үшін қымбат.

"Бірінші кіру - бірінші шығу" стратегиясы процесс беттерінің кадрларын циклдік буфер ретінде қарастырады, одан беттерді циклдік түрде алып тастайды. Бұл стратегияны жүзеге асыру үшін қажет нәрсе-бұл процесс парақтарының кадрларынан циклді түрде өтетін көрсеткіш. Осылайша, бұл алмастыру стратегиясын жүзеге асырудағы қарапайым әдістердің бірі. Оның жұмысының қисыны-негізгі жадтағы бет басқаларға қарағанда ұзағырақ ауыстырылады. Алайда, бұл бет әрдайым сирек қолданылмайды; көбінесе бағдарлама деректердің немесе кодтың кейбір аймақтарын қарқынды қолданады.

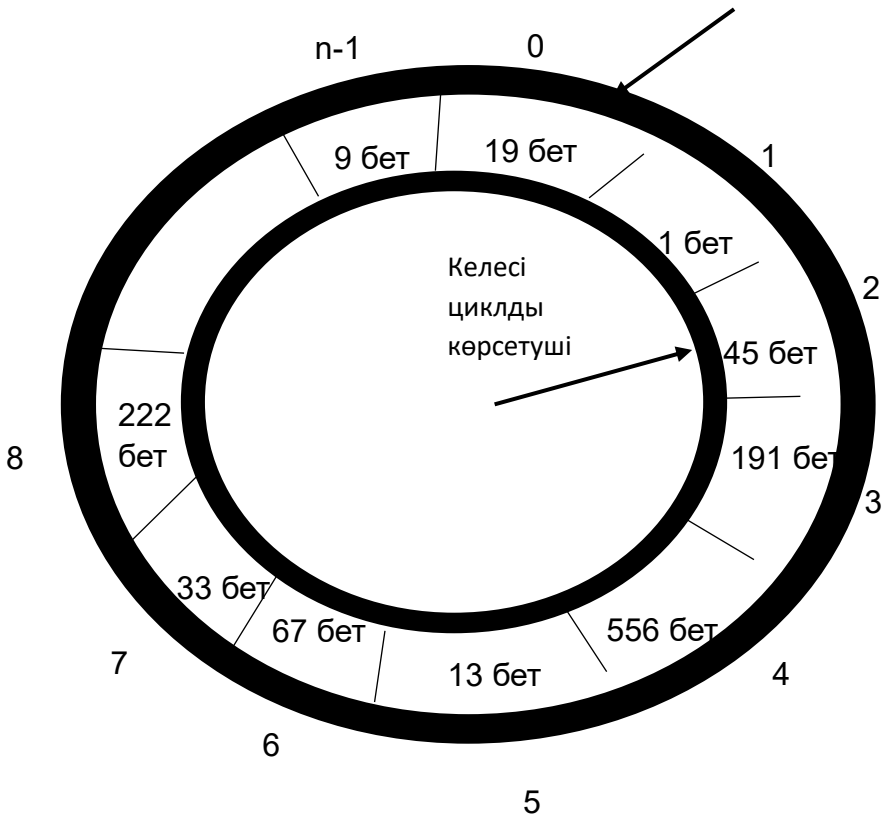
Сағаттық стратегияның қарапайым схемасында әр кадрға пайдалану биті деп аталатын бір қосымша бит қосылады. Бет кадрға алғаш жүктелген кезде, пайдалану биті 1-ге тең орнатылады. Беттің болмауына байланысты үзілісті тудырған бетке келесі қоңыраулар кезінде бұл бит 1-ге тең болады. Ауыстыру алгоритмі жұмыс істеген кезде, ауыстыруға үміткер болып табылатын көптеген кадрлар көрсеткішпен байланысты циклдік буфер ретінде қарастырылады. Бетті ауыстырған кезде көрсеткіш буфердегі келесі кадрға ауысады. Бетті ауыстыру уақыты келгенде, амалдық жүйе 0-ге тең болатын жақтауды іздеу үшін буферді сканерлейді. Іздеу процесінде 1-ге тең пайдалану биті бар кадр пайда болған кезде, ол 0-ге қалпына келтіріледі. Нөлдік битпен бірінші қарсы жақтау ауыстыру үшін таңдалады. Егер барлық жақтауларда 1-ге тең пайдалану биті болса, көрсеткіш толық шеңбер жасайды және осы кадрдағы бетті ауыстыру арқылы бастапқы позицияға оралады. Көріп отырғандай, бұл стратегия "бірінші болып кірді - бірінші болып шықты" стратегиясына ұқсас: бірақ белгіленген пайдалану биті бар кадрларды алгоритм өткізіп жіберетіндігімен ерекшеленеді. Бет жақтауының буфері шеңбер түрінде ұсынылған, онда стратегия атауы пайда болды.

Суретте. 9.7 сағаттық стратегияны қолданудың қарапайым мысалы келтірілген. Ауыстыру үшін негізгі жадтың N кадрлары қол жетімді, олар циклдік буфер түрінде ұсынылған. Екінші жадтан жүктелетін буфердегі бетті 727 бетпен алмастырмас бұрын, буфер көрсеткіші 2 бетін қамтитын 45 кадрға нұсқайды.

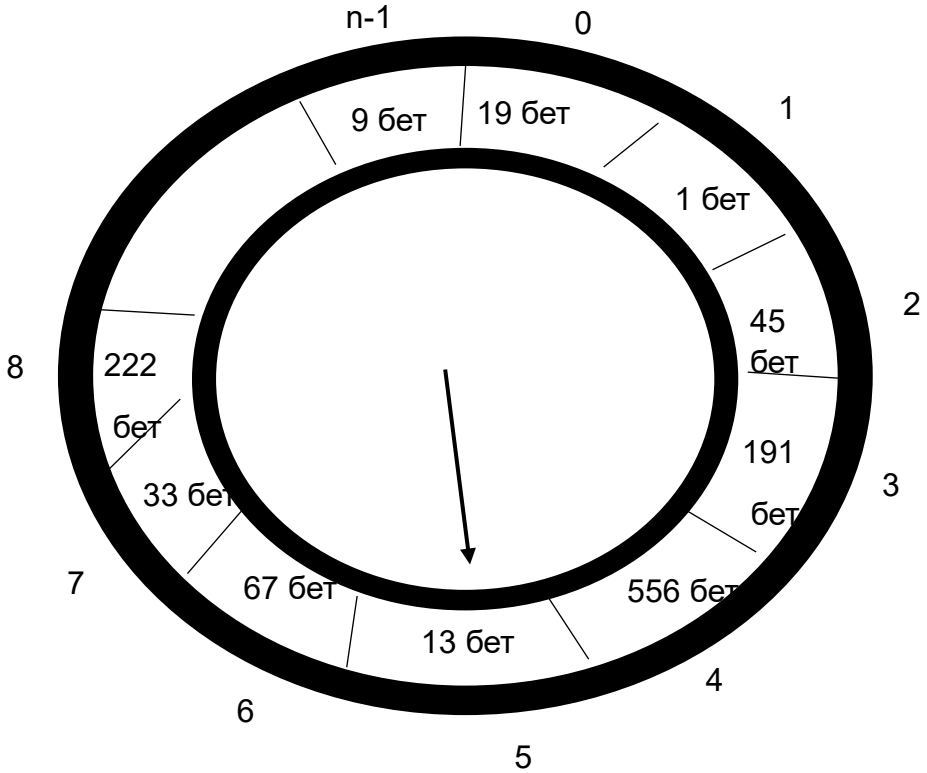
Енді сағат алгоритмін орындауға кірісейік. 2-кадрдағы 45-бетті пайдалану биті 1-ге тең болғандықтан, бұл бет ауыстырылмайды; оның орнына пайдалану биті қалпына келтіріліп, көрсеткіш келесі кадрға ауысады. Сондай-ақ, 191-бет 3-кадрдан ауыстырылмайды; алгоритмге сәйкес оны пайдалану биті қалпына келтіріледі. Келесі кадрда (4 нөмірі) бетті пайдалану биті 0-ге тең. Осылайша, 556 беті негізгі жадқа жүктелген 727 парағымен ауыстырылады, оның қолданылуы 1-ге тең. Әрі қарай, буфер көрсеткіші 5 кадрға өтеді және алгоритмнің орындалуы осымен аяқталады.

Сағат алгоритмінің әрекеті суретте көрсетілген. 9.6. Жұлдызша тиісті бетті қолдану биті 1-ге тең, ал көрсеткі меңзердің ағымдағы орнын көрсетеді.

Суретте. 9.8 осы бөлімде қарастырылған төрт алгоритмді салыстырған [16] эксперимент нәтижелерін көрсетеді; процеске бөлінген кадрлардың саны үнемі болады деп болжанады.

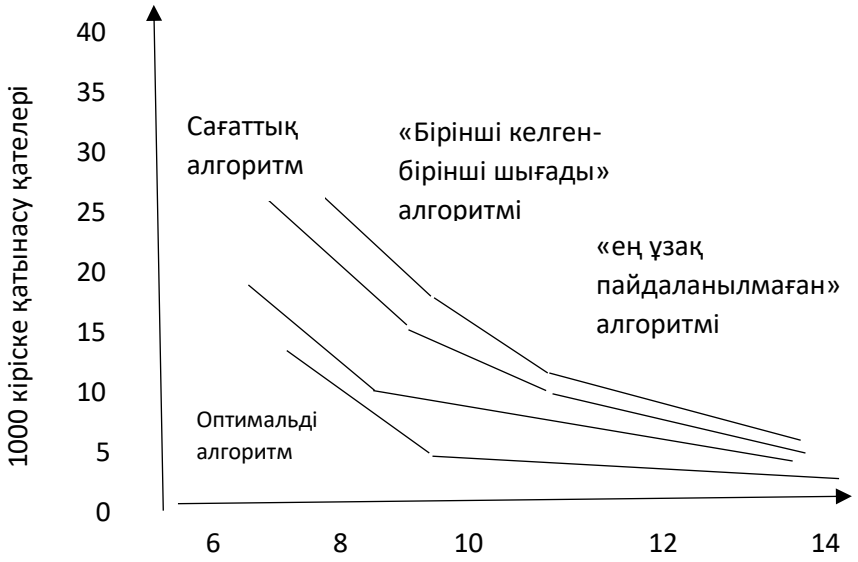


а) Бетті ауыстырар алдындағы буфердің күйі



б) Бетті ауыстырғаннан кейін буфер күйі

Сурет 9.7. Сағат алгоритмінің мысалы



Сурет. 9.8. Беттерді ауыстырудың әртүрлі стратегияларын салыстыру

Резиденттік жиынтығын басқару

Іс жүзінде кез-келген уақытта негізгі жадта орналасқан процестің бөлігі процестің тұрақты жиынтығы болып табылады.

Виртуалды Парақ жадын пайдалану кезінде амалдық жүйе қанша бетті жүктеу керек, яғни белгілі бір процеске қанша жад бөлінетіні туралы шешім қабылдауы керек. Мұнда бірқатар факторлар рөл атқарады.

- Процесске жад неғұрлым аз бөлінсе, соғұрлым көп процестер бір уақытта негізгі жадта болуы мүмкін.
- Сонымен, негізгі жадта орналасқан процесс беттерінің салыстырмалы түрде аз санымен, локализация принципіне қарамастан, парақтың болмауына байланысты үзілістердің жиілігі айтарлықтай үлкен болады.
- Белгілі бір шектеуден кейін негізгі жадты локальдық принципке сәйкес белгілі бір процеске қосымша бөлу парақтың болмауына байланысты үзілістердің пайда болу жиілігінің айтарлықтай төмендеуіне әкелмейді.

Осы факторларды ескере отырып, қазіргі операциялық жүйелерде стратегиялардың екі түрі қолданылады. Белгіленген бөлу стратегиясы процесті жүзеге асырылатын негізгі жад кадрларының белгіленген санына бөледі. Бұл мөлшер жүктеу кезінде анықталады және процестің түріне немесе бағдарламашының немесе жүйелік әкімшінің нұсқауларына сүйене отырып анықталуы мүмкін.

Айнымалы тарату стратегиясы процесс барысында бөлінген бет кадрларының санын өзгертуге мүмкіндік береді. Ең дұрысы, бұл процесс парақтың болмауына байланысты көптеген үзілістерден зардап шегеді (бұл процесс үшін локальдылық принципі нашар орындалады), парақтардың қосымша кадрлары ерекшеленеді; және, керісінше, мұндай үзілістер салыстырмалы түрде аз болатын процесс бар (бұл процестің жергілікті мінез-құлқы жеткілікті жақсы екенін көрсетеді). бұл үзілістердің жиілігін аздап

арттырады деген болжаммен кадрлар алынды. Айнымалы тарату стратегиясы әлдеқайда күшті болып көрінеді, бірақ бұл тәсілдің қиындықтары Операциялық жүйе процестердің мінез-құлқын бақылауы керек. Бұл белгілі бір платформаның аппараттық мүмкіндіктеріне байланысты өте жоғары үстеме шығындарға әкеледі.

Тазалау стратегиясы

Тазарту стратегиясы іріктеу стратегиясына қарама-қайшы. Оның міндеті-өзгертілген бетті екінші жадқа жазу керек сәтті анықтау. Оның екі негізгі әдісі-сұраныс бойынша тазарту және алдын-ала тазарту. Сұраныс бойынша тазалау кезінде Парақ ауыстыру үшін таңдалған кезде ғана екінші жадқа жазылады. Алдын ала тазалау өзгертілген беттерді олар түсірген кадрлар қажет болғанға дейін жазады, сондықтан бұл беттер бүкіл пакеттерде жазылуы мүмкін.

Жақсартылған тәсіл келесі стратегияны қабылдауға мүмкіндік беретін беттерді буферлеуді қамтиды: тек ауыстырылатын беттерді тазарту, сонымен қатар тазалау және ауыстыру операцияларын бөлу. Беттерді буферлеуді пайдаланған кезде ауыстырылатын беттер екі тізімде болуы мүмкін: өзгертілген және өзгертілмеген беттер. Өзгертілген тізімдегі беттер мезгіл-мезгіл пакеттермен жазылып, өзгертілмегендер тізіміне берілуі мүмкін. Өзгертілген беттер тізіміндегі бет оған кірген кезде одан жойылуы мүмкін немесе оның жақтауына жаңа бетті жүктеген кезде жоғалуы мүмкін.

Жүктеуді басқару

Жүктеуді басқару-бұл негізгі жадта сақталатын процестердің санын анықтау. Жүктеуді басқару стратегиясы тиімді жұмыс істейтін жадты басқару жүйесінің маңызды бөлігі болып табылады. Егер резиденттер бір уақытта бірнеше процесті ғана жүргізсе, онда барлық процестер бұғатталатын жағдай жиі туындайды және жүйе свопингті жүзеге асыруға артық уақыт жұмсауға мәжбүр болады. Екінші жағынан, егер сіз негізгі жадта көптеген процестерді орналастырсаңыз, онда орташа алғанда, әр процестің резиденттік жиынының мөлшері өте аз болады, бұл айналым қатесінің тым жиі пайда болуына және жүйенің өткізу қабілетінің төмендеуіне әкеледі.

Осылайша, виртуалды жадты басқару схемасы бағдарламалық және аппараттық қолдауды қажет етеді. Процессор ұсынатын аппараттық қолдау виртуалды адрестерді физикалық адрестерге динамикалық түрлендіруді және негізгі жадта адрестік бет немесе сегмент болмаған кезде үзіліс жасауды қамтиды. Бұл үзіліс жадты басқару бағдарламалық жасақтамасымен өңделеді.

ҚОЛДАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Garg, R.; Verma, G.Operating Systems [OP]: An Introduction - Softcover Publisher: Mercury Learning & Information, 2017. 290 p.
2. https://gifer.com/ru/7h0m
3. https://3dnews.ru/1034959
4. Darrell Hajek, Cesar Herrera, Flor Narciso Principles of Operating Systems. Independently Published (24 April 2020) 176 pages.
5. Andrew S. Tanenbaum and Herbert Bos. Modern Operating Systems. 4/E.1136 pages, Pearson India, 2016.
6. Silberschatz Abraham, Galvin Peter Baer and Gadne Greg. Operating system concepts.
7. Amdahl GM (1967) Validity of the single-processor approach to achieve large scale computing capabilities. AFIPS Joint Spring Conference Proceedings 30 (Atlantic City, NJ, Apr. 18–20), AFIPS Press, Reston VA, pp 483–485.
8. https://studfile.net/.
9. https://habr.com/ru/post/40227/.
10. wikimedia.org
11. wordpress.com
12. blackandwhitecomputer.blog
13. http://www-inst.eecs.berkeley.edu/~n252/paper/Amdahl.pdf.
14. encyclopedia2.thefreedictionary.com
15. linustechtips.com
16. youtube.com/watch?v=w3K1JklY6D4