

## Дәріс 10 Енгізу/шығаруды басқару

### 10.1. Енгізу/шығару құрылғылары

### 10.2. Енгізу/шығару функцияларын ұйымдастыру

Енгізу/шығару функцияларының эволюциясы

Жадқа тікелей қол жеткізу

### 10.3. Пайдалану сметасын жобалау мәселелері

Дизайн мақсаттары

Енгізу/шығару функцияларының логикалық құрылымы

### 10.4. I/O буферлеу

Қос буфер

Дөңгелек буфер

Буферлеуді пайдалану

### 10.5. Дискіні жоспарлау

Диск өнімділігі опциялары

Іздеу уақыты

Айналуға байланысты кідіріс

Тасымалдау уақыты

Уақытты салыстыру

Дискіні жоспарлау стратегиялары

FIFO

Басымдықтар: соңғы кірген, бірінші шыққан

SSTF

SCAN

C-SCAN

N-қадамды SCAN және FSCAN

### 10.6. RAID

RAID 0

RAID 0 және жаппай деректерді тасымалдау

RAID 0 және жоғары енгізу/шығару сұраныс жылдамдығы

RAID 1

RAID 2

RAID 3

Артықшылық

Өнімділік

RAID 4

RAID 5

RAID 6

### 10.7. Дискінің кәші

Даму мәселелері

Өнімділік мәселелері

### 10.1. Енгізу/шығару құрылғылары

Компьютерлік жүйелерде енгізу/шығару операцияларын қамтамасыз ететін сыртқы құрылғыларды үш топқа бөлуге болады.

1. Пайдаланушымен жұмыс. Компьютер пайдаланушысымен байланысу үшін қолданылады. Мысал ретінде дисплейден, пернетақтадан, тінтуірден тұратын принтерлер мен бейне терминалдарды келтіруге болады.

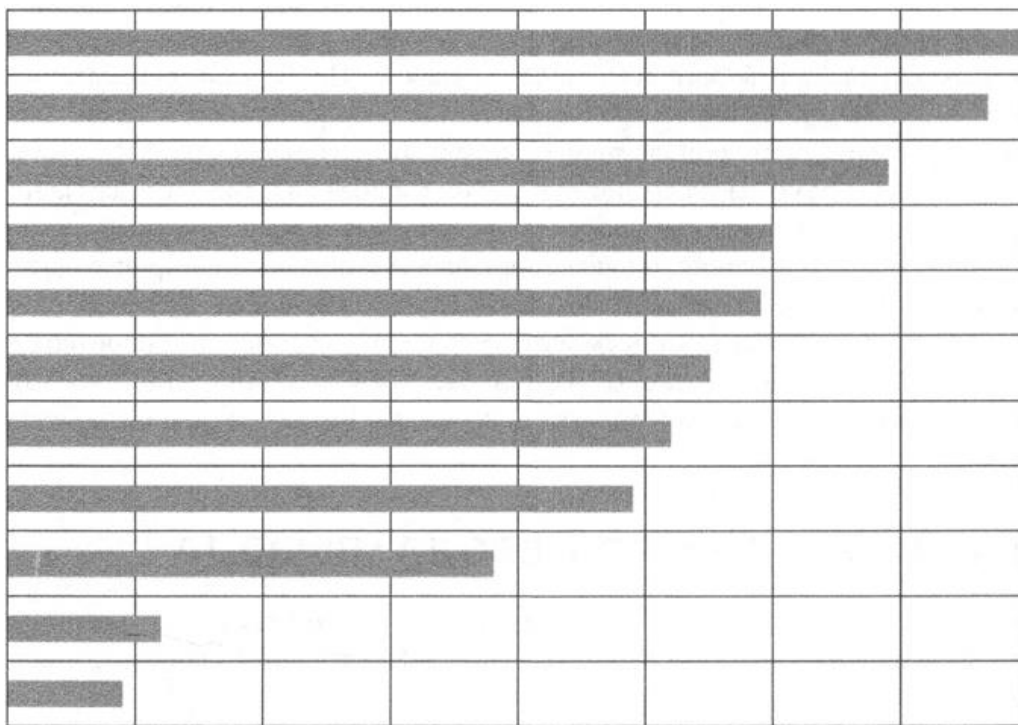
2. Компьютермен жұмыс. Электрондық жабдықпен байланысу үшін қолданылады. Оларға диск құрылғылары, USB кілттері, сенсорлар, контроллерлер және түрлендіргіштер жатады.

3. Коммуникациялар. Қашықтағы құрылғылармен байланысу үшін қолданылады. Оларға модемдер мен сандық желі драйверлері жатады.

Енгізу/шығару құрылғыларының арасында айтарлықтай айырмашылықтар бар:

– **Тасымалдау жылдамдығы** . Мәліметтерді беру жылдамдығы бірнеше шама ретімен өзгеруі мүмкін (10.1-сурет).

– **Қолданба** . Құрылғы қолдайтын әрбір әрекет операциялық жүйенің бағдарламалық жасақтамасы мен стратегияларына әсер етеді.



Деректерді беру жылдамдығы, б/сек

Сурет. 10.1. Әдеттегі енгізу/шығару құрылғылары үшін деректерді беру жылдамдығы

– **Басқарудың күрделілігі**. Басқару интерфейстеріндегі айырмашылықтардың операциялық жүйеге әсері енгізу/шығару контроллерлерінің күрделілігінің артуымен азайтылады.

– **Мәліметтерді тасымалдау бірліктері**. Деректерді байт немесе таңбалар ағыны ретінде (мысалы, терминалды енгізу/шығару кезінде) және блоктарда (мысалы, дискінің енгізу/шығаруын орындау кезінде) тасымалдауға болады.

– **Деректерді көрсету**. Әртүрлі құрылғылар әртүрлі таңба кодтаулары мен паритеттерін қоса алғанда, әртүрлі деректерді кодтау схемаларын пайдаланады.

– **Қате жағдайлары**. Қателердің сипаты, оларды хабарлау тәсілі, олардың салдары және ықтимал жауаптар бір құрылғыдан екіншісіне күрт өзгереді.

Мұндай әртүрлілік, шын мәнінде, операциялық жүйе тұрғысынан да, пайдаланушы процестері тұрғысынан да енгізу-шығару мәселесіне бірыңғай және дәйекті тәсілді әзірлеу мүмкін еместігіне әкеледі.

## 10.2. Енгізу/шығару функцияларын ұйымдастыру

Үш негізгі енгізу/шығару әдісі бар.

1. Бағдарламаланатын енгізу/шығару. Процессор енгізу-шығару контроллеріне қажетті командаларды жібереді; содан кейін процесс енгізу-шығару операциясының аяқталуын күтетін күйде болады.

2. Үзілістермен басқарылатын енгізу/шығару. Процессор процестің атынан енгізу-шығару контроллеріне қажетті командаларды жібереді. Бұл жағдайда екі мүмкіндік бар. Егер процесстен енгізу/шығару нұсқауы блокталмаған болса, процессор енгізу/шығару нұсқауын шығарған процестің нұсқауларын орындауды жалғастырады. Егер процестің енгізу-шығару командасы блокталатын болса, онда процессор орындайтын келесі пәрмен операциялық жүйеден келеді, ол ағымдағы процесті блокталған күйге келтіреді және басқа процесті іске қосуды жоспарлайды.

3. Жадқа тікелей қол жеткізу (Direct Memory Access DMA). DMA модулі негізгі жады мен енгізу/шығару контроллері арасындағы деректер алмасуды басқарады. Процессор деректер блогын жадқа тікелей қол жеткізу модуліне тасымалдауға сұраныс жібереді, ал үзіліс тек бүкіл деректер блогын тасымалдағаннан кейін пайда болады.

### **Енгізу/шығару функцияларының эволюциясы**

Енгізу-шығару құрылғыларының функционалдық мүмкіндіктерінің даму кезеңдерін келесідей сипаттауға болады.

1. Процессор перифериялық құрылғыны тікелей басқарады. Мұны процессорлар басқаратын қарапайым құрылғылардан көруге болады.

2. Құрылғыға контроллер немесе енгізу/шығару модулі қосылған. Процессор бағдарламаланатын енгізу/шығару қызметін үзіліссіз пайдаланады. Бұл кезеңде процессор сыртқы құрылғы интерфейстерінің нақты бөлшектерінен біршама бөлектенеді.

3. Процессор үзіліссіз бағдарламаланатын кіріс-шығыс қолданады. Нәтижесінде, процессорға кіріс-шығыс операцияларының орындалуын күтуге уақыт жұмсаудың қажеті жоқ, бұл өнімділіктің жоғарылауына әкеледі.

4. Енгізу/шығару модулі DMA көмегімен жадпен тікелей жұмыс істеу мүмкіндігін алады. Процессорды пайдаланбай деректер блогын жадқа немесе одан шығаруға болады (деректерді тасымалдаудың басталуы мен аяқталуын қоспағанда).

5. Енгізу-шығару модулі жетілдірілуде және шығаруға арналған арнайы командалар жүйесі бар жеке процессорға айналады. Орталық процессор енгізу/шығару процессорына негізгі жадта орналасқан енгізу/шығару бағдарламасын орындауға нұсқау береді. Енгізу-шығару процессоры орталық процессордың қатысуынсыз тиісті нұсқауларды алады және орындайды. Бұл процессорға енгізу/шығару функцияларының орындалу ретін анықтауға және бүкіл реттілік орындалғанда ғана үзілуіне мүмкіндік береді.

6. Енгізу-шығару модулінің өзінің жергілікті жады бар және шын мәнінде жеке компьютер болып табылады. Осы архитектураның көмегімен бірнеше енгізу-шығару құрылғыларын ең аз CPU араласуымен басқаруға болады. Әдетте, мұндай архитектура интерактивті терминалдармен байланысты басқару үшін қолданылады. Енгізу/шығару процессоры терминалдарды басқарумен байланысты көптеген тапсырмаларды орындайды.

Егер біз жоғарыда сипатталған енгізу-шығару құрылғыларының даму жолын ұстанатын болсақ, процессордың енгізу-шығару функцияларына араласуы азайып бара жатқанын байқай аламыз. Орталық процессор енгізу/шығарумен байланысты тапсырмалардан көбірек босатылады, нәтижесінде жалпы өнімділік артады. 5 және 6-кезеңдер енгізу-шығару құрылғысының тұжырымдамасының өзгеруін көрсетеді - бұдан былай ол бағдарламаны өз бетінше орындауға қабілетті.

Терминологияға назар аударыңыз. Параграфтарда сипатталған барлық модульдер үшін. 4-6, тікелей қол жеткізу термині өте қолайлы, өйткені олардың әрқайсысы негізгі жадты енгізу-шығару модулі арқылы тікелей басқаруды пайдаланады. 5-тармақта

сипатталған енгізу/шығару модулі жиі енгізу/шығару арнасы деп аталады, ал б-тармақта сипатталған модуль енгізу/шығару процессоры деп аталады.

### Жадқа тікелей қол жеткізу

10.2-Суретте жалпыланған DMA логикалық диаграммасы. Жадқа тікелей қол жеткізу құрылғысы процессордың функцияларын қайталауға, атап айтқанда, процессордан жүйені басқаруды қабылдауға қабілетті. Бұл мүмкіндік оған жүйелік шинаға деректерді беру үшін қажет - жадқа да, одан да.

Сурет. 10.2. Жадқа тікелей қол жеткізудің блок-схемасы

Процессорға деректер блогын оқу немесе жазу қажет болғанда, ол келесі ақпаратты жібере отырып, DMA модуліне сұраныс жасайды.

– Операциялардың қайсысы – оқу немесе жазу – сұралады. Осыған байланысты - оқуды немесе жазуды басқару сызығы процессор мен DMA модулі арасында пайдаланылады.

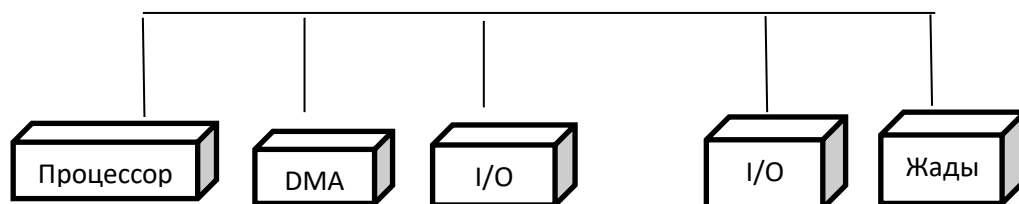
– Деректерге қосылған пайдаланудағы енгізу/шығару құрылғысының мекенжайы.

– DMA модулінің мекенжай регистрінде сақталған оқылатын (немесе жазылатын) жад аймағының бастапқы мекенжайы.

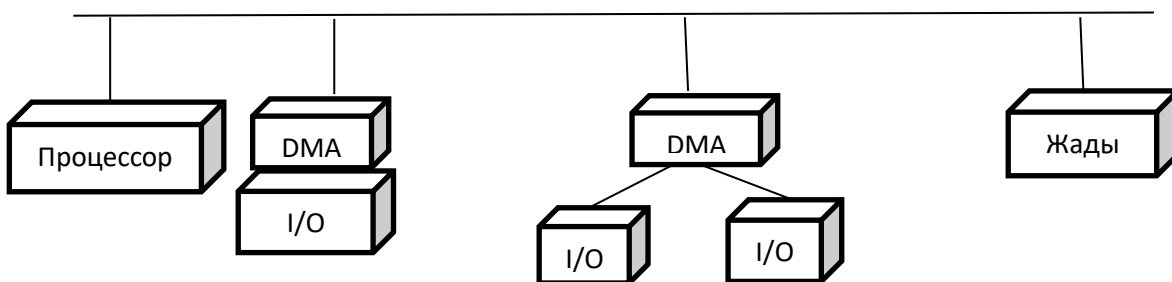
– Қанша сөзді оқу немесе жазу керек. Бұл мән DMA модулінің деректер есептегіш регистрінде сақталады.

Осыдан кейін процессор енгізу-шығару операциясын басқаруды DMA модуліне бере отырып, басқа тапсырмамен жұмысын жалғастырады. Өз кезегінде, DMA модулі процессорды айналып өтіп, деректердің бүкіл блогын тікелей жадқа тасымалдайды (немесе одан деректерді оқиды). Деректерді тасымалдау аяқталғаннан кейін DMA модулі процессорға үзу сигналын жібереді. Осылайша, процессор бұл процеске деректерді берудің басында және соңында ғана қосылады.

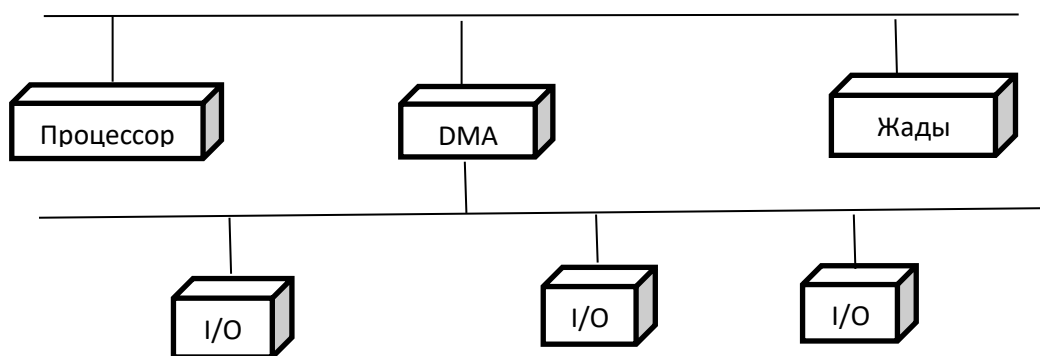
DMA конфигурациясын әртүрлі жолдармен жасауға болады; олардың кейбіреулері суретте көрсетілген. 10.3.



а) жеке шина, жеке DMA



б) DMA-I/O біріктірілген бір шина



с) енгізу/шығару шинасы

Сурет. 10.3. Баламалы тікелей жадқа кіру конфигурациялары

### 10.3. Операциялық жүйені жобалау мәселелері

#### Дизайн мақсаттары

Енгізу-шығару құралдарын жобалау кезінде әзірлеушілер екі мақсатты басшылыққа алады: жоғары тиімділік пен әмбебаптыққа қол жеткізу. Тиімділік өте маңызды, өйткені енгізу-шығару операциялары көбінесе компьютерлік жүйеде кідірістердің пайда болуына ықпал етеді. Бұл мәселені шешудің бір әдісі-көп функциялы режим. Процессордың жоғары жүктемесін сақтау үшін жадқа дайын процестерді жүктейтін swap қолданылады, бірақ ол Енгізу-шығару операциясынан басқа ештеңе емес. Осылайша, операциялық жүйені құру кезіндегі басты назар енгізу-шығару операцияларының тиімді схемасын іздеуге бағытталған.

Тағы бір маңызды мақсат – жан-жақтылық. Енгізу/шығару құрылғыларымен жұмыс істеуді жеңілдету және қателердің болу мүмкіндігін азайту үшін әртүрлі құрылғыларды бірдей басқару мүмкіндігін алған жөн. Бұл пайдаланушы процестерімен де, операциялық жүйемен де енгізу/шығару құрылғыларын басқаруға қатысты. Енгізу/шығару функциясын жобалауға арналған модульдік тәсіл енгізу/шығару құрылғысының төменгі деңгейлі мәліметтерінің көпшілігін жасырады, осылайша пайдаланушы процестері мен жоғары деңгейдегі операциялық жүйе операциялары құрылғыға оқу және жазу сияқты жалпы функциялар тұрғысынан ғана қол жеткізе алады. ашу және жабу, құлыптау және құлыпты ашу.

#### Енгізу/шығару функцияларының логикалық құрылымы

Операциялық жүйе функцияларын олардың күрделілігіне, сипаттамалық уақыт шкаласына және абстракция деңгейіне қарай бөлу керек. Енгізу/шығару ұйымының мәліметтері құрылғы түріне және оның қолданбасына байланысты. 10.4-суретте ең маңызды үш логикалық құрылым көрсетілген.

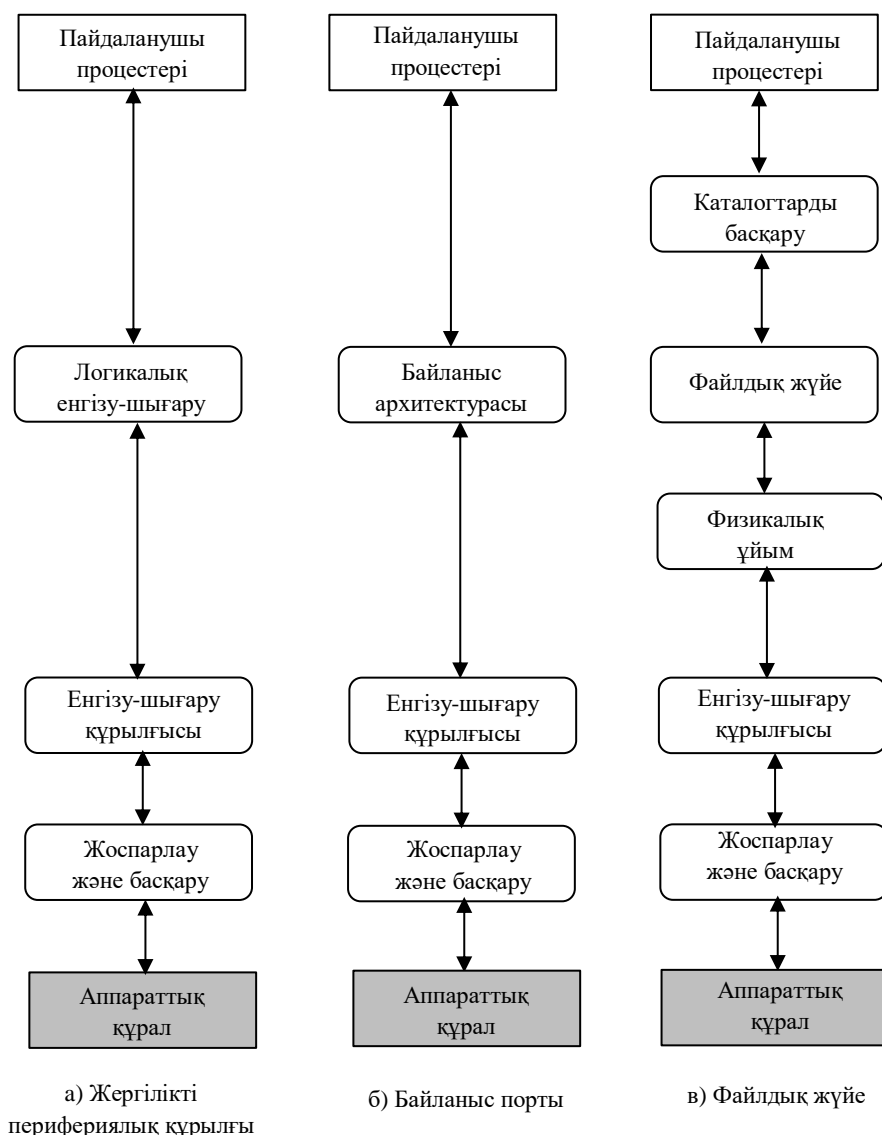
Құрылғы байт немесе жазбалар ағыны арқылы байланысқан қарапайым жағдайды қарастырайық (1 0.4, *a- сурет* ). Бұл жағдайда деңгейлер келесідей болады.

– **Логикалық енгізу/шығару.** Логикалық енгізу/шығару модулі құрылғыны логикалық ресурс сияқты қарастырады және құрылғыны нақты басқару мәліметтеріне мән бермейді.

– **Енгізу/шығару құрылғысы.** Сұралған операциялар мен деректер енгізу/шығару нұсқауларының, арналарды басқару командаларының және контроллер командаларының сәйкес реттіліктеріне түрлендіріледі.

– **Жоспарлау және бақылау.** Бұл деңгейде енгізу-шығару операцияларын нақты кезекке қою және жоспарлау, сонымен қатар операциялардың орындалуын бақылау орын алады. Осылайша, бұл деңгейде үзілістермен жұмыс, құрылғының күйі туралы ақпаратты қабылдау және беру жүзеге асырылады. Бұл енгізу/шығару контроллерімен, демек

құрылғының аппараттық құралдарымен тікелей әрекеттесетін бағдарламалық жасақтама деңгейі.



Сурет. 10.4. Енгізу/шығару ұйымының моделі

Байланыс құрылғылары үшін енгізу-шығару құрылымы (сурет 10.4, б) жоғарыда сипатталғанмен бірдей дерлік көрінеді. Негізгі айырмашылығы - логикалық енгізу-шығару модулі байланыс архитектурасымен ауыстырылады, ол өз кезегінде бірнеше қабаттардан тұруы мүмкін.

Суретте. 10.4с суретінде файлдық жүйені қолдайтын екінші сақтау құрылғысына енгізу/шығаруды басқарудың типтік құрылымы көрсетілген. Мұнда біз бұрын кездеспеген үш деңгей бар.

**1. Каталогты басқару.** Бұл деңгей файл дескрипторын немесе индекс кестесін пайдалана отырып, таңбалық файл атауларын тікелей немесе жанама түрде файл идентификаторларына түрлендіреді. Бұл деңгей сонымен қатар оларды қосу, жою немесе қайта ұйымдастыру сияқты файлдар каталогындағы пайдаланушы әрекеттерімен байланысты.

**2. Файлдық жүйе.** Бұл деңгей файлдардың логикалық құрылымымен және ашу, жабу, оқу және жазу сияқты пайдаланушы көрсеткен операциялармен айналысады. Сонымен қатар, қол жеткізу құқықтарын басқару да осы деңгейде орын алады.

3. **Физикалық ұйым.** Виртуалды Жад мекен-жайларын сегментация мен бетті ұйымдастыруды ескере отырып, негізгі жадтың физикалық мекен-жайларына айналдыру керек, сонымен қатар файлдар мен жазбаларға логикалық сілтемелер осы құрылғының жолдары мен секторларының физикалық құрылымын ескере отырып, белгілі бір екінші реттік сақтау құрылғысының физикалық мекен-жайларына түрлендірілуі керек. Сол деңгейде сыртқы жад кеңістігі мен негізгі жад буферлерінің таралуын жалпы басқару жүзеге асырылады.

#### 10.4. Енгізу/шығаруды буферлеу

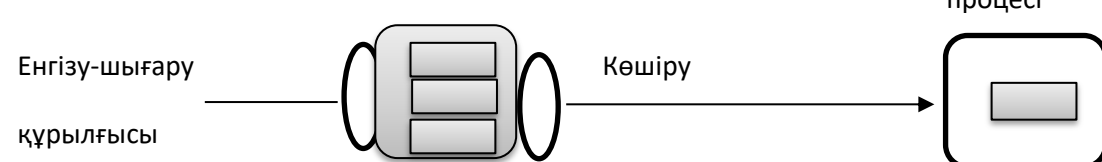
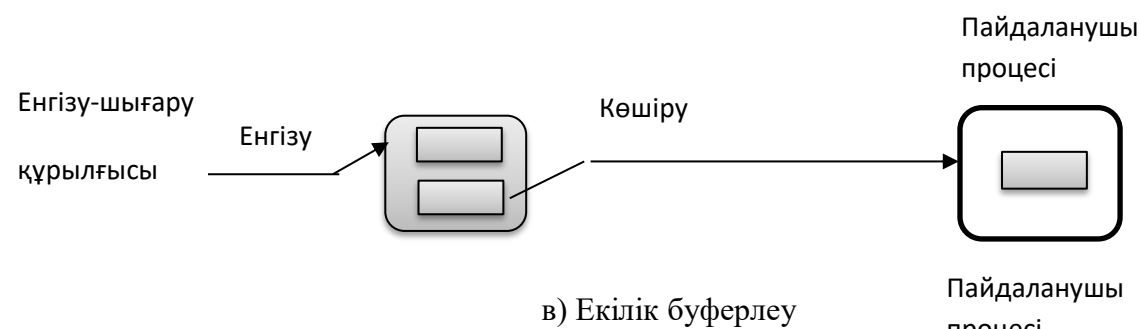
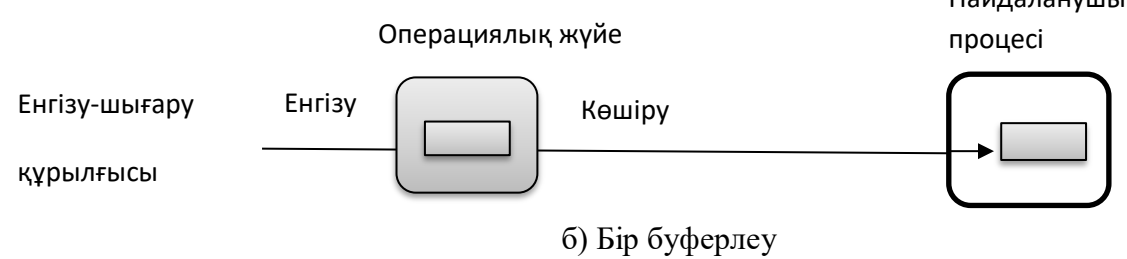
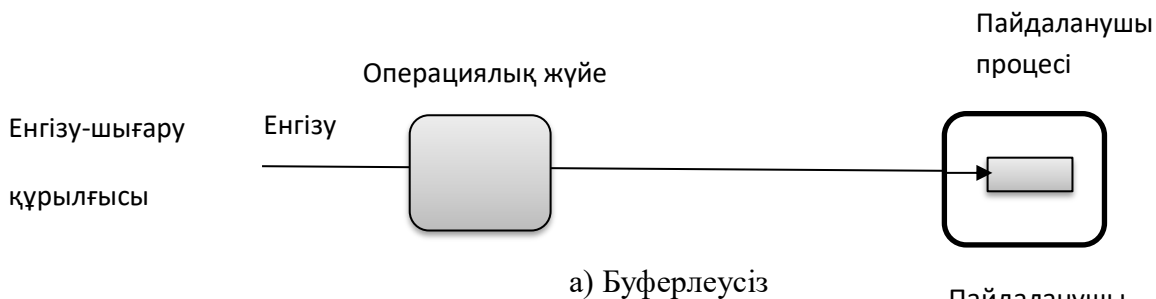
Үстеме шығындарды азайту және тиімділікті арттыру үшін кейде нақты сұранысқа дейін деректерді алдын ала оқу (және деректерді нақты сұраудан сәл кейінірек жазу) ыңғайлы (10.5-сурет, *a*). Бұл технология **буферлеу** ретінде белгілі.

Өртүрлі буферлеу әдістерін қарастырған кезде енгізу/шығару құрылғыларының екі түрі бар екенін есте ұстаған жөн: блокқа бағытталған және ағынға бағытталған. Блок-бағытталған құрылғылар ақпаратты блоктарда сақтайды, әдетте белгіленген өлшемді және блок бойынша деректерді тасымалдауды жүзеге асырады. Әдетте, блок нөмірін пайдаланып деректерге сілтеме жасай аласыз. Дискілер мен USB кілттері блок негізіндегі енгізу/шығару құрылғыларының мысалдары болып табылады. Жіпті қорғау құрылғылары блоктық құрылымсыз байттардың құрылымданбаған ағындары түрінде деректерді тасымалдауды жүзеге асырады. Бұл құрылғылар тобына терминалдар, принтерлер, байланыс порттары, тінтуір және басқа меңзегіш құрылғылар, сондай-ақ сыртқы жад құрылғылары болып табылмайтын көптеген құрылғылар кіреді.

Операциялық жүйеден қолдаудың ең қарапайым түрі бір буфер болып табылады (сурет 10.5, *b*). Пайдаланушы процесі енгізу/шығару сұранысын орындаған кезде - операциялық жүйе оған негізгі жадтың жүйелік жағында буфер тағайындайды.

Блокқа бағытталған құрылғыларға арналған жалғыз буфер схемасын келесідей сипаттауға болады. Біріншіден, енгізу жүйелік буферге жіберіледі. Ол аяқталғаннан кейін процесс блокты пайдаланушы кеңістігіне жылжытады және бірден келесі блоктың сұрауын жасайды. Бұл процедура алдын-ала оқу немесе алдын-ала енгізу деп аталады; ол уақыт өте келе бұл блок талап етіледі деген болжаммен орындалады. Тапсырмалардың көптеген түрлері үшін бұл әдіс көп жағдайда жақсы жұмыс істейді, өйткені деректерге қол жеткізу әдетте дәйекті түрде жүзеге асырылады (өңдеу тізбегі аяқталғаннан кейін ғана блокты оқу артық болады).

Жалпы жағдайда бұл тәсіл буферлеудің жоқтығымен салыстырғанда өнімділіктің жоғарылауын қамтамасыз етеді. Пайдаланушы процесі келесі блок оқылып жатқанда бір деректер блогын өңдей алады. Ұқсас пайымдаулар блокқа бағытталған шығысқа қатысты. Деректер құрылғыға тасымалданатын болса, ол алдымен пайдаланушы кеңістігінен жүйелік буферге көшіріледі, содан кейін ол құрылғыға жазылады. Бұл жағдайда деректер жүйелік буферге тасымалданғаннан кейін шығару процесі бірден жұмысын жалғастыра алады.





### г) Циклдік буферлеу

#### Сурет. 10.5. Енгізу/шығару буферлеу схемалары (енгізу)

Бір буфер схемасы ағынды-бағытталған енгізу-шығару жолымен немесе байтпен қолданылуы мүмкін

##### Қос буфер

Бір буферлік схеманы екі жүйелік буферді қолдану арқылы жақсартуға болады (сурет. 10.5, в). Енді процесс деректерді бір буферге жібереді (немесе одан оқиды), ал Операциялық жүйе екіншісін босатады (немесе толтырады). Бұл технология Қос Буферлеу немесе буферлік алмасу деп аталады. Ағынға бағытталған енгізу кезінде біз тағы екі балама жұмыс режиміне жүгінеміз. Жолды шығару кезінде процесті тоқтата тұру қажеттілігі келесі жолды шығару кезінде екі буфер бос болмаған жағдайда ғана туындайды. Байттық операцияларда Қос буфердің Қос ұзындықтағы бір буферден артықшылығы жоқ. Екі жағдайда да өндіруші/ тұтынушы моделі қолданылады.

##### Дөңгелек буфер

Қос буферлік Схема кіріс-шығыс құрылғысы мен процесс арасындағы деректер ағынын теңестіруге арналған. Егер біз белгілі бір процестің жұмысына қызығушылық танытатын болсақ, онда, ең алдымен, енгізу/шығару операциялары оның жұмысына кедергі келтірмеуі керек. Егер процесс жиі кіріс немесе шығыс болса, қос Буферлеу жеткіліксіз болуы мүмкін. Көбінесе бұл жағдайда буферлер санының көбеюі мәселені шешуге көмектеседі. Екіден көп буферді қолданған кезде схема циклдік Буферлеу деп аталады (сурет. 10.5, г). Онда әрбір жеке буфер циклдік буфер модулі болып табылады.

##### Буферлеуді пайдалану

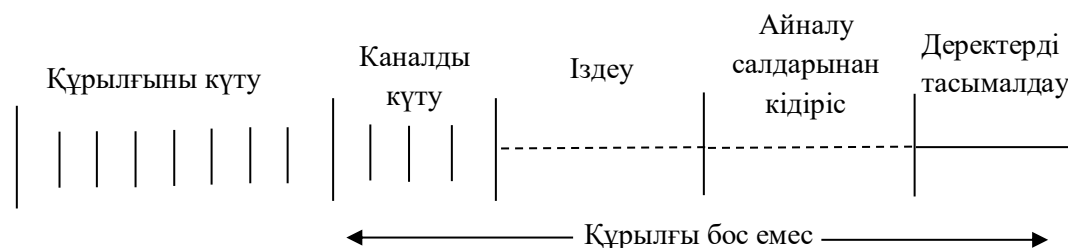
Буферлеу – енгізу/шығару сұрауларының санындағы өсулерді тегістеуге арналған әдіс. Енгізу/шығару сұраулары бар әртүрлі процестер және құрылғылардың бірдей әртүрлілігі бар көп тапсырмалы ортада буферлеу операциялық жүйенің тұтастай өнімділігін де, жеке процестердің өнімділігін де арттыра алатын құрал болып табылады.

#### 10.5. Дискіні жоспарлау

Процессорлар мен негізгі жадтың жылдамдығын арттыру дискіге қол жеткізу жылдамдығымен салыстырғанда үлкен қорғасымен жүзеге асырылады. Сондықтан дискілік жүйенің өнімділігі өмірлік маңызды мәселе болып табылады және оны жақсарту жолдарын іздеуге көптеген зерттеу жұмыстары жүргізілді.

##### Диск өнімділігі опциялары

Дискінің енгізу/шығару әрекетінің нақты мәліметтері компьютер жүйесіне , операциялық жүйеге, енгізу/шығару арнасының сипатына және диск контроллерінің аппараттық құралына байланысты. Дискінің енгізу-шығару құрылғысы арқылы деректерді берудің жалпыланған уақыт диаграммасы Суретте көрсетілген. 10.6.



Сурет. 10.6. Дискінің уақытының диаграммасы

Диск жұмыс істеп тұрған кезде оның айналу жылдамдығы тұрақты болады. Оқу немесе жазу үшін бас ізделетін жолдың үстінде, дәлірек айтқанда, осы жолдағы ізделетін сектордың басында болуы керек. Жолды таңдау процедурасы басын жылжытуды (жылжымалы басы бар жүйеде) немесе қалаған басын электронды таңдауды (бекітілген

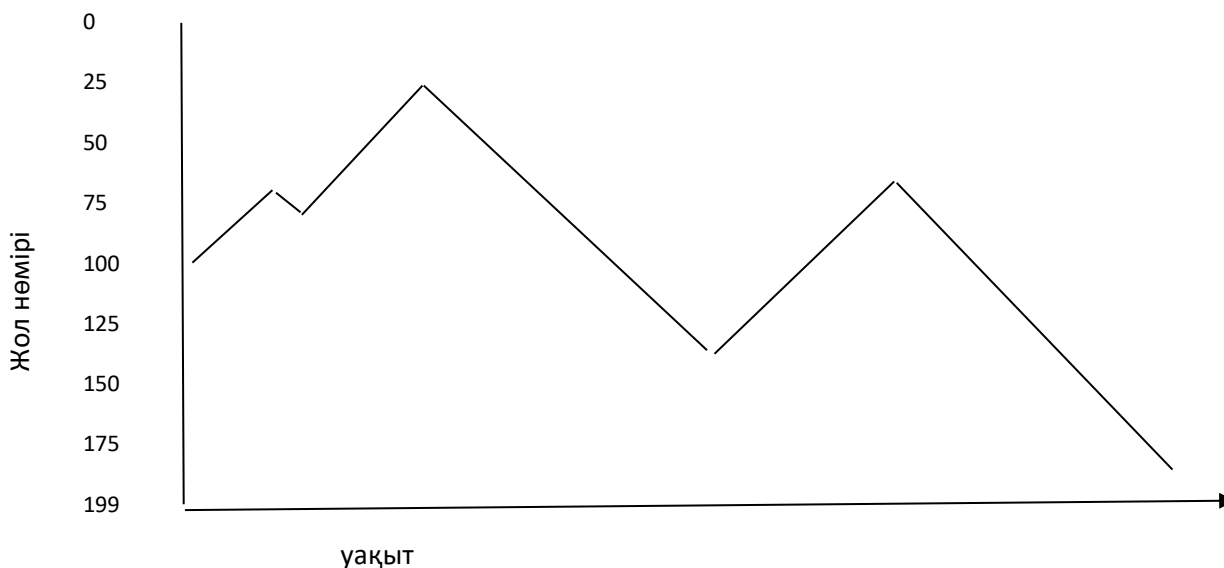
басы бар жүйеде) камтиды. Жылжымалы бастары бар жүйеде іздеу уақыты деп аталатын уақыт жолдың үстіне бастың орналасуына жұмсалады. Қалай болғанда да, жолды таңдағаннан кейін, диск контроллері қалаған сектордың басы басына жеткен сәтті күтеді. Сектордың басталуына жету үшін қажетті уақыт айналу немесе айналу уақытын күту уақытына байланысты кідіріс уақыты ретінде белгілі. Айналдыру уақытына байланысты іздеу уақыты мен кідіріс уақытының қосындысы-қол жеткізу уақыты – оқу немесе жазу үшін орналастыру уақыты. Басы қалаған позицияға енгеннен кейін, оқу немесе жазу операциясы жасалады, ол сектордың бастың астындағы қозғалысы кезінде жүзеге асырылады – бұл Енгізу-шығару операциясы кезінде деректерді тікелей беру болып табылады. Деректерді беру үшін қажет уақыт беру уақыты деп аталады.

Бұған қоса, әдетте дискінің енгізу/шығаруымен байланысты бірқатар басқа кешігулер бар. Процесс енгізу/шығару сұрауын орындаған кезде, соңғысы құрылғының қолжетімді болуын күтіп тұрған кезекке қойылуы керек. Осыдан кейін құрылғы процеске тағайындалады. Құрылғы енгізу/шығару арналарын басқа дискілермен ортақ пайдаланса, арна қолжетімді болғанша күту қажет болуы мүмкін. Осыдан кейін ғана бұрын талқыланған дискіге тікелей қол жеткізу жүзеге асырылады.

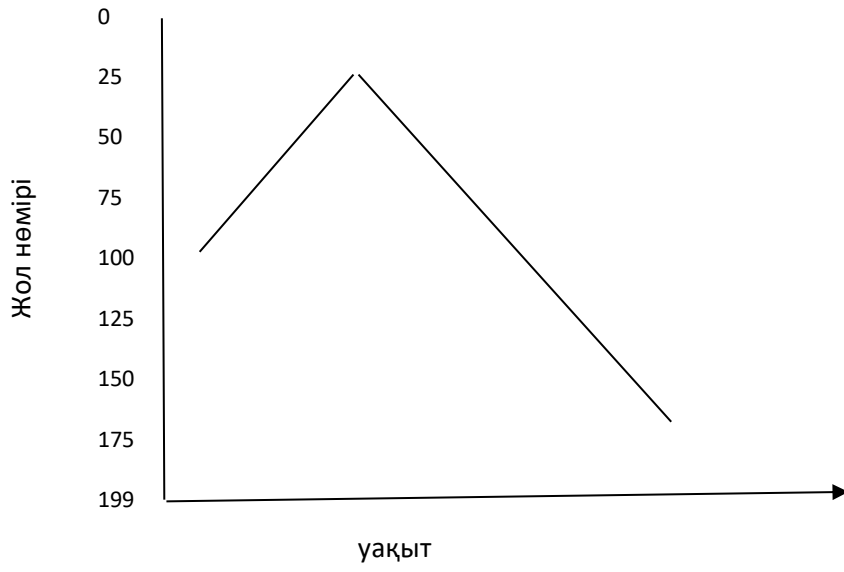
### Дискіні жоспарлау стратегиялары

Операциялық жүйе әрбір енгізу/шығару құрылғысы үшін сұраулар кезегін сақтайтын көп тапсырмалы ортадағы әдеттегі жағдайды қарастырыңыз. Сәйкесінше, бір дискінің кезегіндегі әртүрлі процестерден енгізу/шығару сұраныстарының белгілі саны (оқу немесе жазу) болады. Кезектен сұрауларды кездейсоқ таңдасаңыз, онда сіз іздеген тректер кездейсоқ ретпен болады деп күтуіңіз керек және бұл өте нашар өнімділікке әкеледі.

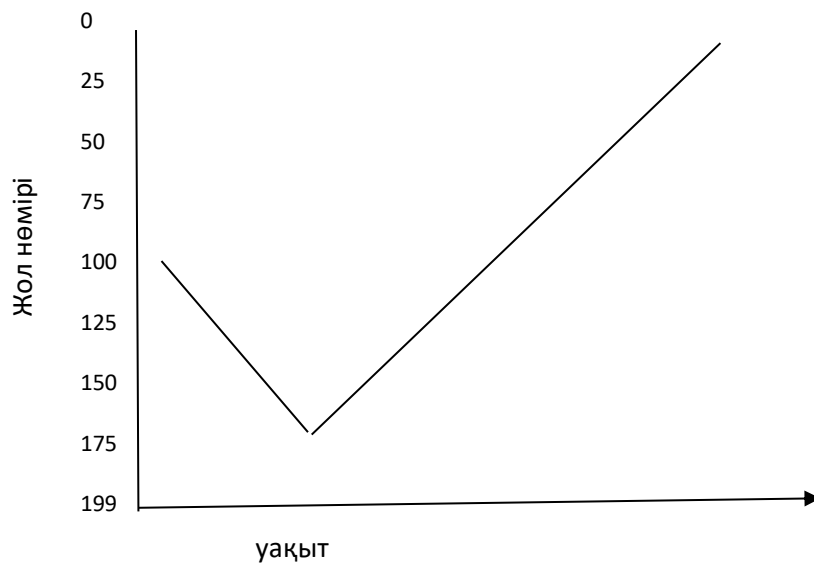
Суретте. 10.7 әр түрлі жоспарлау алгоритмдерінің өнімділігі енгізу/шығару сұрауларының мысалы үшін салыстырылады. Тік ось диск жолдарына сәйкес келеді; көлденең ось уақытқа сәйкес келеді немесе өткен жолдардың санына сәйкес келеді. Бұл суретте дискінің басы бастапқыда 100 жолда орналасқан деп болжанады; дискіде 200 трек бар, ал дискіге сұрау кезегінде кездейсоқ сұраулар болады (сұратылған жолдар, дискіні жоспарлаушыға кіру реті бойынша - 55, 58, 39, 18, 90, 160, 150, 38, 184). Кестеде. 10.1 сандық нәтижелер келтірілген.



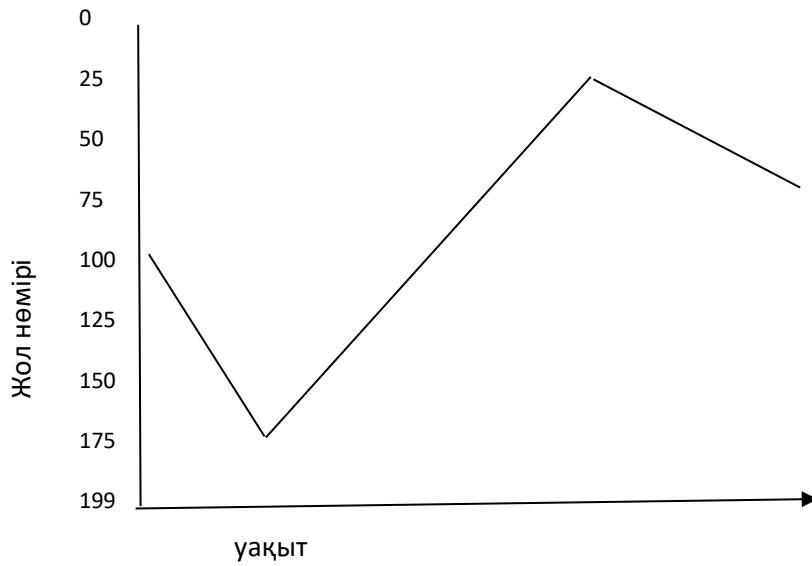
a)FIFO



б)SSTF



в)SCAN



г)C-SCAN

**Сурет. 10.7.** Дискіні жоспарлау алгоритмдерін салыстыру

а)FIFO (бастапқы жол 100)		б)SSTF (бастапқы жол 100)		в)SCAN (бастапқы жол 100;жол нөмірін ұлғайту бағытына бағыт)		г)C-SCAN (бастапқы жол 100; жол нөмірін азайту бағытына бағыт )	
Келесі жол	Қиылықан жолдар саны	Келесі жол	Қиылықан жолдар саны	Келесі жол	Қиылысқан жолдар саны	Келесі жол	Қиылысқан жолдар саны
55	45	90	10	150	50	150	50
58	3	58	32	160	10	160	10
29	19	55	3	184	24	184	24
18	21	39	16	90	94	18	166
90	72	38	1	58	32	38	20
160	70	18	20	55	3	39	1
150	10	150	132	39	16	55	16
38	112	160	10	38	1	58	3
184	146	184	24	18	20	90	32
Іздеудің орташа ұзындығы	55,3	Іздеудің орташа ұзындығы	27,5	Іздеудің орташа ұзындығы	27,8	Іздеудің орташа ұзындығы	35,8

10.1-кесте. Дискіні жоспарлау алгоритмдерін салыстыру

**FIFO**

Жоспарлаудың қарапайым түрі – "бірінші келді-бірінші шықты" (FIFO) жоспарлау, онда кезек элементтері кезекпен өңделеді. Бұл стратегияның артықшылығы әділеттілікте, өйткені әр сұрау орындалады және барлық сұраулар қабылдау тәртібімен орындалады. 10.2-кестеде дискіні жоспарлау алгоритмдерінің сипаттамалары мен ерекшеліктері көрсетілген.

10.2-кесте. Дискіні жоспарлау алгоритмдері

Аты	Сипаттама	Ескертпелер
Сұраныс көзіне сәйкес таңдау		
Кездейсоқ	кездейсоқ жоспарлау	Талдау және модельдеу үшін
FIFO	«Бірінші кірген, бірінші шыққан»	Ең бейтарап әдіс
PRI	Процесс басымдығы	Диск сұрау кезегі сырттан басқарылады
LIFO	«Соңғы кірген, бірінші шыққан»	Жергілікті жерді және ресурстарды пайдалануды барынша арттыру
Сұраныс мазмұнына сәйкес таңдау		

SSTF	Ең қысқа қызмет көрсету уақытын таңдау	Жоғары пайдалану, төмен кезек
SCAN	Дискіде алға және артқа жылжу	Жақсырақ қызмет көрсету
C-SCAN	Жылдам қайтарумен бір бағытты қозғалыс	Қызмет көрсетудің төмен өзгергіштігі
N-қадамы-SCAN	Бір бумадағы N жазбамен SCAN	Қызмет көрсету кепілдігі
FSCAN	N - қадам - SCAN , мұндағы N - SCAN циклінің басындағы кезек өлшемі	Жүктемеге сезімтал

### **Басымдықтар**

Жүйеде басымдықты (PRI) қолдана отырып, жоспарлауды басқару дискіні басқарудың бағдарламалық жасақтамасына қатысты сыртқы болып табылады. Көбінесе қысқа пакеттік тапсырмалар, сондай-ақ интерактивті тапсырмалар ұзақ есептеулерді қажет ететін ұзақ тапсырмаларға қарағанда жоғары басымдыққа ие болады. Бұл схема жүйеде көптеген қысқа тапсырмаларды тез аяқтауға мүмкіндік береді және қысқа жауап беру уақытын қамтамасыз етеді.

### **Соңғы кірген - бірінші шыққан**

Бір таңқаларлығы, алғашқы соңғы сұранысты орындау стратегиясының артықшылықтары бар. Транзакцияларды өңдеу жүйелерінде құрылғыны соңғы пайдаланушыға ұсынған кезде сериялық файл көрсеткішінің аз ғана қозғалысы орындалуы керек. Жергілікті артықшылықтарды пайдалану өткізу қабілетін арттыруға және кезек ұзындығын азайтуға мүмкіндік береді. Өкінішке орай, егер дискідегі жүктеме үлкен болса, процесті ашығудың айқын мүмкіндігі бар.

Талқыланған үш жоспарлау стратегиясы - FIFO, PRI және LIFO - тек кезек немесе сұрау процесінің атрибуттарына негізделген. Дегенмен, егер жоспарлаушы ағымдағы жолды білсе, онда сұраудың мазмұнына негізделген жоспарлау стратегиясын пайдалануға болады.

### **SSTF**

Ең аз қызмет көрсету уақытын таңдау стратегиясы (Shortest Service Time First-SSTF) ағымдағы күйден бастарды ең аз жылжытуды қажет ететін дискіні енгізу-шығару сұранысын таңдау болып табылады. Әрине, іздеудің минималды уақытын үнемі таңдау барлық қозғалыстарда іздеудің орташа уақыты минималды болатынына кепілдік бермейді; дегенмен, бұл стратегия диск жүйесінің FIFO-мен салыстырғанда жақсы жұмысын қамтамасыз етеді.

### **SCAN**

SCAN алгоритмін қолданған кезде бастың қозғалысы тек бір бағытта жүреді, таңдалған бағытқа сәйкес келетін сұраныстарды қанағаттандырады. Таңдалған бағытта соңғы жолға жеткеннен кейін (немесе мүмкін сұраулар таусылған кезде) бағыт керісінше өзгереді. Бұл соңғы нақтылау кейде көрініс стратегиясы деп аталады. Содан кейін қызмет бағыты керісінше өзгереді және сканерлеу қарама-қарсы бағытта жалғасады, қайтадан барлық сұраныстарды рет-ретімен жинайды.

### **C-SCAN**

C-SCAN (Циклдік сканерлеу) стратегиясы сканерлеуді тек бір бағытта шектейді. Берілген бағыттағы соңғы жол табылғанда, басы дискінің қарама-қарсы ұшына оралады және сканерлеу қайтадан басталады.

### **N - қадам - SCAN**

N-қадам-SCAN стратегиясы диск сұрау кезегін N ұзындықтағы ішкі кезектерге бөледі. Әрбір ішкі кезек SCAN стратегиясын пайдаланып бір әрекетте өңделеді. Кезек өңделіп жатқандықтан, жаңа сұраулар басқа кезекке қосылуы мүмкін. Ағымдағы

сканерлеудің соңында N сұраудан аз болса, олардың барлығы келесі сканерлеу циклінде өңделеді.

### **FSCAN**

FSCAN - екі ішкі кезекті пайдаланатын стратегия. Сканерлеу басталған кезде барлық сұраулар кезектердің бірінде болады; ал екіншісі бос қалады. Бірінші кезекті сканерлеу кезінде барлық жаңа сұраулар екінші кезекке түседі. Осылайша, жаңа кезектерге қызмет көрсету барлық ескі сұраулар өңделгенше кейінге қалдырылады.

### **10.6. RAID**

Жоғарыда айтылғандай, қайталама сақтау өнімділігінің өсуі процессорлар мен негізгі жадтан әлдеқайда артта қалады. Бұл сәйкессіздік жалпы өнімділік деңгейін жоғарылату кезінде диск жүйесіне ерекше назар аударуға мәжбүр етеді.

Басқа салалардағы сияқты, бірнеше құрылғыларды қатар пайдалану арқылы қосымша тиімділікке қол жеткізуге болады. Дискілер жағдайында бұл тәуелсіз және параллель дискілердің массивтерін пайдалануды білдіреді. Бірнеше дискілер болған кезде, рұқсат етілген деректер блогы бірнеше дискілер бойынша таратылса, әртүрлі енгізу/шығару сұраулары қатар өңделуі мүмкін. Сонымен қатар, егер қол жеткізілетін деректер блогы бірнеше дискілерге таралса, тіпті бір енгізу-шығару сұрауы параллельді түрде орындалуы мүмкін.

Бірнеше дискілерді пайдаланған кезде сенімділікті арттыру үшін деректерді ұйымдастырудың және артық қосудың көптеген нұсқалары бар. RAID салалық стандарты бар ( Reundant массив ның Тәуелсіз Дискілер – тәуелсіз дискілердің артық массиві). RAID схемасы 7 деңгейден тұрады - нөлден алтыға дейін. Бұл деңгейлер иерархиялық құрылымға ие емес, бірақ келесі жалпы сипаттамалары бар әртүрлі архитектураларды анықтайды.

1. RAID – операциялық жүйе бір логикалық диск ретінде қарастыратын физикалық дискілердің жиынтығы.

2. Деректер массивтің физикалық дискілеріне таралады.

3. Дискінің артық сыйымдылығы диск ақауы кезінде деректерді қалпына келтіруді қамтамасыз ету үшін паритеттік ақпаратты сақтау үшін пайдаланылады.

Екінші және үшінші сипаттамалар әртүрлі RAID деңгейлері үшін әртүрлі. RAID 0 және 1 деңгейлері үшінші мүмкіндікті мүлде қолдамайды.

RAID термині алғашында Берклидегі Калифорния университетіндегі әзірлеушілер тобының ғылыми есебінде қолданылған [ 186 ]. Презентацияда RAID-тің әртүрлі конфигурациялары мен пайдаланулары, сондай-ақ RAID деңгейлерінің анықтамалары қысқаша талқыланды. Бұл стратегия жоғары тығыздықты дискілерді бірнеше төмен тығыздықты дискілермен ауыстырады және деректерді әртүрлі дискілерден бір уақытта алуға болатындай етіп таратады. Бұл енгізу/шығару тиімділігін айтарлықтай жақсартады және массивтің қосымша сыйымдылығын кеңейтуге мүмкіндік береді.

Ұсынылған технологияның бірегейлігі артықшылықты тиімді пайдалануында. Көбірек дискілердің болуы өнімділікті жақсартады , бірақ сәтсіздікке ұшырау мүмкіндігін арттырады. Осыған байланысты RAID ақаулық жағдайына байланысты жоғалған деректерді қалпына келтіруге мүмкіндік беретін қосымша ақпаратты сақтауды қамтамасыз етеді.

Кестеде. 10.4 барлық жеті деңгейге арналған жылдам нұсқаулықты қамтиды.

#### 10.4-кесте. RAID деңгейлері

Санат	Денг ей	Сипаттама	Дискілер дің қажетті саны	Деректердің қолжетімділігі	Үлкен деректер өткізу қабілеті	Шағын енгізу/шығару сұранысының жылдамдығы
-------	---------	-----------	---------------------------	----------------------------	--------------------------------	--

Сызат	0	Артықшылық жоқ	N	Бір дисктен аз	Өте биік	Өте жоғары оқу және жазу
Рефлексия (айналау)	бір	Рефлексия	2N	RAID 2, 3, 4 және 5-тен көп, бірақ RAID 6 - дан аз	Бірнеше дискіні оқу үшін; жазу үшін бір дискімен салыстыруға болады	Бір дискінің оқу сыйымдылығын екі есе дерлік арттыру; жазу үшін бір дискімен салыстыруға болады
Параллелді қатынас	2	Хамминг кодтарымен артық	N+m	Бір дискіден әлдеқайда жоғары, RAID 3, 4 және 5-пен салыстыруға болады	Барлық аталған баламалардың ішіндегі ең үлкені	Бір дискіден екі есе дерлік көп
	3	Битаралық паритет	N+1	Бір дискіден әлдеқайда жоғары, RAID 2, 4 және 5-пен салыстыруға болады	Барлық аталған баламалардың ішіндегі ең үлкені	Бір дискіден екі есе дерлік көп
Тәуелсіз қол жеткізу	төрт	Жолақты паритет	N+1	Бір дискіден әлдеқайда жоғары, RAID 2, 3 және 5-пен салыстыруға болады	Оқу үшін ол RAID 0-ге ұқсас; бір дискіге қарағанда жазуға әлдеқайда аз	Оқу үшін ол RAID 0-ге ұқсас; бір дискіге қарағанда жазуға әлдеқайда аз
	5	Жолақты блоктармен бөлінген теңдік	N+1	Бір дискіден әлдеқайда жоғары, RAID 2, 3 және 4-пен салыстыруға болады	Оқу үшін ол RAID 0-ге ұқсас; бір дискіден аз жазу үшін	Оқу үшін ол RAID 0-ге ұқсас; жазу үшін, жалпы алғанда, бір дискіден аз
	6	Жолақты блоктары бар қос үлестірілген паритет	N+2	Барлық аталған баламалардың ішіндегі ең үлкені	Оқу үшін ол RAID 0-ге ұқсас; RAID 5 -тен аз жазу үшін	Оқу үшін ол RAID 0-ге ұқсас; RAID 5 -ке қарағанда жазу әлдеқайда аз

N – Деректер дискілерінің саны;  $m \log N$  пропорционал .

### 10.7. Диск кэші

Кэш жады локализация принципіне байланысты жадқа кірудің орташа уақытын қысқартады. Дәл осындай принципті дискіні сақтау үшін де қолдануға болады. Дискінің кэші кейбір диск секторларының мазмұны үшін негізгі жадтағы буфер болып табылады. Егер енгізу/шығару сұранысы белгілі бір секторға қатысты болса, онда ол алдымен кэште осы сектордың болуын тексереді. Егер сектор кэште болса, онда сұраныс кэштен қанағаттандырылады. Әйтпесе, сұралған сектор дискіден кэшке оқылады. Егер бір енгізу-шығару сұрауын орындау үшін деректер блогы қажет болса, онда локализация принципіне сүйене отырып, оның жақын арада қайтадан қажет болуы әбден мүмкін.

## 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](http://wikimedia.org)
11. [wordpress.com](http://wordpress.com)
12. [blackandwhitecomputer.blog](http://blackandwhitecomputer.blog)
13. <http://www-inst.eecs.berkeley.edu/~n252/paper/Amdahl.pdf>.
14. [encyclopedia2.thefreedictionary.com](http://encyclopedia2.thefreedictionary.com)
15. [linustechtips.com](http://linustechtips.com)
16. [youtube.com/watch?v=w3K1JkIY6D4](https://youtube.com/watch?v=w3K1JkIY6D4)