Дәріс 11. Файлдарды басқару

11.1. Файлдар мен файлдық жүйелер
Файл құрылымы
Файлдарды басқару жүйелері
Ф сәулеті
Файлдық жүйе
Файлдарды басқару функциялары
11.2. Файлдарды ұйымдастыру және оларға қол жеткізу
Аралас файл
Сериялық файл
Индекс-сериялық файл
Индекстелген файл
Тікелей қатынау файлы
11.3. Файл каталогтары
Мазмұны
Құрылымы
Атау
11.4. Файлдарды ортақ пайдалану
Қол жеткізу құқықтары
Бір уақытта қол жеткізу
11.5. Екінші жадыны басқару
Файлдарды орналастыру
Алдын-ала және динамикалық орналастыру
Қызмет ету мөлшері
Файлдарды орналастыру әдістері
Бос кеңістікті басқару
Биттік кестелер
Бос бөліктер тізбегі
Индекстеу
Бос блоктардың тізімі
Томдар
Сенімділік

11.1. Файлдар мен файлдық жүйелер

Пайдаланушы тұрғысынан операциялық жүйенің маңызды бөліктерінің бірі-пайдаланушыларға келесі қасиеттері бар файлдар деп аталатын мәліметтер жиынтығын жасауға мүмкіндік беретін файлдық жүйе.

- **Ұзақ мерзімді жүйе**. Файлдар дискіде немесе басқа екінші жадта сақталады және пайдаланушы жүйеден шыққан кезде жойылмайды.
- **Процестерді ортақ пайдалану**. Файлдарда атаулар бар және басқарылатын ортақ пайдалануға мүмкіндік беретін байланысты қол жетімділік құқықтары болуы мүмкін.
- **Құрылымы**. Файлдық жүйеге байланысты файл белгілі бір қосымшалар үшін ыңғайлы ішкі құрылымға ие болуы мүмкін. Сонымен қатар, файлдар иерархиялық немесе файлдар арасындағы қатынасты көрсететін күрделі құрылымға ұйымдастырылуы мүмкін.

Кез-келген файлдық жүйе файл түрінде ұйымдастырылған деректерді сақтау құралдарын ғана емес, сонымен қатар файлдарда орындалатын функциялар жиынтығын да ұсынады. Типтік операцияларға мыналар кіреді:

- **Жасау**. Жаңа файл файлдық жүйеде анықталады және орналастырылады.
- **Жою.** Файл файлдық жүйеден жойылады.

- **Ашу.** Қолданыстағы файл белгілі бір процесс арқылы "ашық" деп жарияланады, бұл процестің файл үстінде әртүрлі әрекеттерді орындауына мүмкіндік береді.
- **Жабу**. Процесске арналған файл жабылады, содан кейін процесс файлды қайта ашқанша файлдағы әрекеттерді орындай алмайды.
- **Оқу.** Процесс файлдағы барлық деректерді немесе олардың бір бөлігін оқиды.
- **Жазба**. Процесс файл өлшемін арттыратын жаңа деректерді қосу немесе файлдағы бар деректер элементтерінің мәндерін өзгерту арқылы файлды жаңартады.

Әдетте файлдық жүйе файлға қатысты атрибуттар жиынтығын қолдайды. Оларға файл иесі, оны жасау уақыты, соңғы өзгерту уақыты және кіру рұқсаттары кіреді.

Файл құрылымы

Файлдарды зерттеу кезінде төрт негізгі термин қолданылады.

- Өріс
- Жазу
- Файлды алу
- Деректер базасы.

Өріс (field) - деректердің негізгі элементі. Жеке өрісте жалғыз мән бар. Өріс деректердің ұзындығы мен түрімен сипатталады (мысалы, ASCII жолы, ондық сан және т.б.). Файл құрылымына байланысты өрістер тұрақты немесе өзгермелі ұзындыққа ие болуы мүмкін. Соңғы жағдайда өріс көбінесе екі немесе үш ішкі өрістен тұрады: нақты мән, өріс атауы және кейде өріс ұзындығы.

Жазба (record) - бұл кейбір қолданбалы бағдарламамен бірлік ретінде өңделетін өзара байланысты өрістер жиынтығы. Құрылымына байланысты жазбалар тұрақты немесе өзгермелі ұзындықта болуы мүмкін.

Файл (файл) - біртекті жазбалар жиынтығы. Пайдаланушылар мен қосымшалар файлды тұтастай қарастырады және оған жүгіну оның аты бойынша жүзеге асырылады. Файлдарды жасауға және жоюға болады; олардың әрқайсысының өзіндік ерекше атауы бар. Кіру шектеулері әдетте файл деңгейінде жүзеге асырылады.

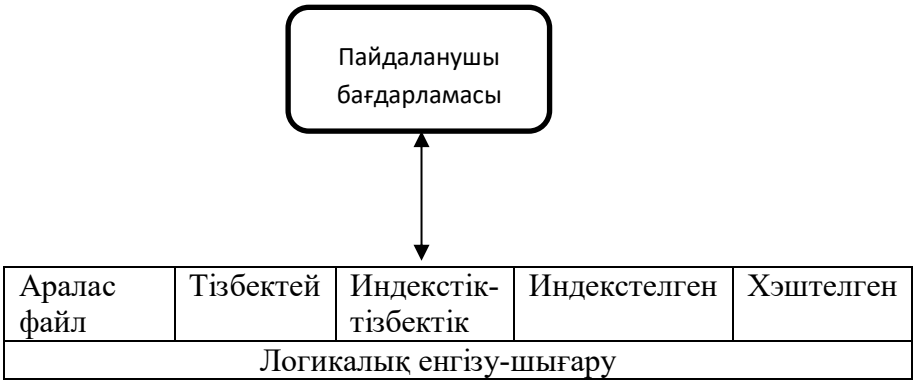
Деректер базасы (database) - өзара байланысты мәліметтер жиынтығы. Деректер базасының маңызды аспектілеріне деректер арасындағы қатынастар айқын көрінеді, ал Дерекқордың өзі көптеген қосымшаларды пайдалану үшін арнайы жасалған. Деректер базасы бір немесе бірнеше файлдардан тұрады. Әдетте операциялық жүйеге тәуелсіз жеке дерекқорды басқару жүйесі (ДҚБЖ) бар, дегенмен ол әрдайым дерлік кейбір басқару бағдарламаларын қолданады файл пайдаланушылар мен қосымшалар, келесі операцияларға қолдау көрсетіледі.

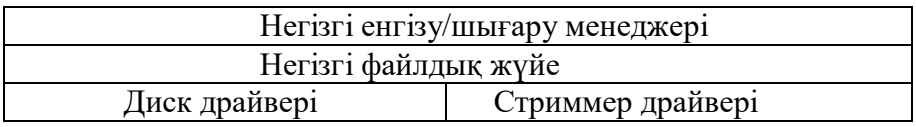
Файлдарды басқару жүйелері

Файлдарды басқару жүйесі пайдаланушылар мен файлдарды қолдану кезінде қызметтік функцияларды қамтамасыз ететін бағдарламалық жасақтама болып табылады. Әдетте файлдармен жұмыс істеудің жалғыз әдісі-файлдарды басқару жүйесін пайдалану.

Файлдық жүйенің архитектурасы

11.1-суретте файлдарды басқару бағдарламалық жасақтамасын ұйымдастырудың типтік сызбасы көрсетілген.





Сурет 11.1. Файлдық жүйенің бағдарламалық жасақтамасының архитектурасы

Әр түрлі жүйелер әртүрлі ұйымдастырылған. Төменгі деңгейде құрылғы драйверлері тікелей перифериялық құрылғылармен немесе олардың контроллерлерімен немесе арналарымен өзара әрекеттеседі. Құрылғы драйвері құрылғының кіріс-шығыс операцияларына және кіріс-шығыс сұрауының аяқталуын өңдеуге жауап береді. Файлдық операцияларда магниттік таспадағы диск пен диск жетегі бақыланатын құрылғылар болып табылады. Құрылғы драйверлері әдетте операциялық жүйенің бөлігі ретінде қарастырылады.

Келесі деңгей негізгі файлдық жүйе немесе физикалық енгізу-шығару деңгейі деп аталады. Бұл компьютерлік жүйенің айналасымен негізгі интерфейс. Ол диск жүйесімен, магниттік таспамен немесе басқа тасымалдаушымен алмасатын деректер блоктарымен жұмыс істейді, сондықтан ол жедел жадқа блоктарды орналастыру және буферлеумен байланысты. Бұл деңгейде деректер блогының мазмұнымен немесе файл құрылымымен жұмыс жасалмайды. Негізгі файлдық жүйе көбінесе операциялық жүйенің бөлігі ретінде қарастырылады.

Негізгі енгізу-шығару супервайзері (диспетчері) файлды енгізу-шығаруды бастау және аяқтау үшін жауап береді. Бұл деңгейде енгізу-шығару құрылғысына, файлдарды жоспарлауға және күйге байланысты басқару құрылымдары қолданылады. Негізгі енгізу-шығару бақылаушысы таңдалған файлға негізделген файлды енгізу-шығару операциясы орындалатын құрылғыны таңдайды. Негізгі енгізу-шығару супервайзері операциялық жүйенің бөлігі болып табылады.

Логикалық енгізу-шығару пайдаланушылар мен қосымшаларға жазбаларға қол жеткізуге мүмкіндік береді. Осылайша, негізгі файлдық жүйе деректер блоктарымен, ал логикалық енгізу-шығару модулі файлдық жазбалармен жұмыс істейді. Логикалық енгізу-шығару жазбаларды енгізу-шығару бойынша жалпы мақсаттағы мүмкіндіктерді қамтамасыз етеді және файлдар туралы ақпаратты қолдайды.

Файлдық жүйенің пайдаланушыға ең жақын деңгейі көбінесе кіру әдісі деп аталады. Ол қолданбалар мен файлдық жүйелер мен деректерді қамтитын құрылғылар арасындағы стандартты интерфейсті қамтамасыз етеді. Әртүрлі қол жеткізу әдістері әртүрлі файл құрылымдарын және деректерге қол жеткізу мен өңдеудің әртүрлі жолдарын көрсетеді.

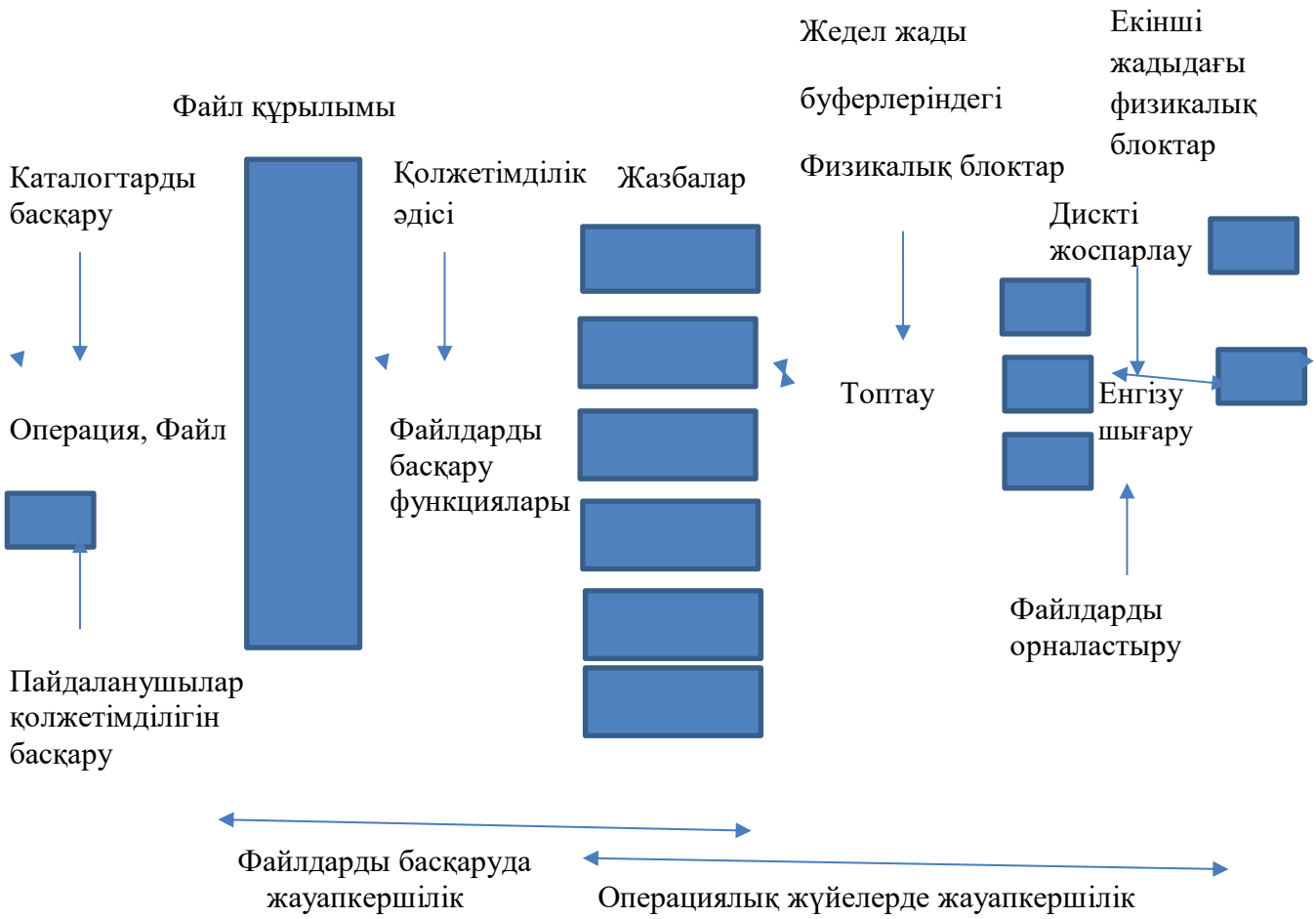
Файлдарды басқару функциялары

Файлдық жүйенің функциялар 11.2-суретте көрсетілген. Осы диаграмманы солдан оңға қарай қарастыру керек. Пайдаланушылар мен қосымшалар файлдарды құруға және жоюға, сондай-ақ олармен операцияларды орындауға арналған командалар арқылы файлдық жүйемен өзара әрекеттеседі. Кез-келген әрекетті жасамас бұрын, файлдық жүйе таңдалған файлды анықтап, оның орналасқан жерін анықтауы керек. Бұл файлдардың орналасқан жерін және олардың сипаттамаларын сипаттауға қызмет ететін белгілі бір типтегі каталогтарды қолдануды қажет етеді. Сонымен қатар, көптеген бірлескен жүйелер пайдаланушыларға қол жеткізуді басқаруды қамтиды. Пайдаланушы немесе бағдарлама файлдардың үстінен орындай алатын негізгі операциялар жазбалар деңгейінде орындалады. Пайдаланушы немесе қолданбалы бағдарлама тұрғысынан файл жазбаларды ұйымдастыратын дәйекті құрылымға ие. Сондықтан пайдаланушы командаларын нақты файлды басқару командаларына түрлендіру үшін берілген файлдың құрылымына сәйкес келетін қол жеткізу әдісі қолданылуы керек.

Пайдаланушылар мен қосымшалар жазбалармен жұмыс жасаса, төмен деңгейлі енгізу-шығару блоктармен орындалады. Сондықтан файл жазбаларын шығару үшін топтастыру керек және енгізгеннен кейін топтастыру керек. Блоктық файлдарды енгізу-шығаруды қолдау үшін бірнеше мүмкіндіктер қажет. Сыртқы сақтау құрылғыларында сақтау басқарылуы керек. Басқару мүмкіндіктеріне сыртқы құрылғыларда сақталған кезде

файлдарды бос блоктарға бөлу және жаңа файлдарды жасау және бұрыннан барларын құру үшін қандай блоктар бар екенін білу үшін бос жадты басқару кіреді.

11.2-суретте файлдарды басқару жүйесі мен операциялық жүйе арасында бөлу ұсынылған; қиылысу нүктесі-жазбаларды өңдеу. Мұндай бөлу дұрыс, өйткені әртүрлі жүйелер әртүрлі тәсілдерді қолданады.



Сурет 11.2. Файлдарды басқару элементтері

11.2. Файлдарды ұйымдастыру және оларға қол жеткізу

Файлды ұйымдастыру әдісін таңдағанда бірнеше өлшемдерді ескеру қажет.

- Кіру жылдамдығы.
- Жаңарту жеңілдігі.
- Сақтау үнемділігі.
- Техникалық қызмет көрсету оңай.
- Сенімділік.

Бұл критерийлердің салыстырмалы басымдық деңгейі файлмен жұмыс істейтін қосымшаларға байланысты. Төменде файлды ұйымдастырудың бес әдісі берілген.

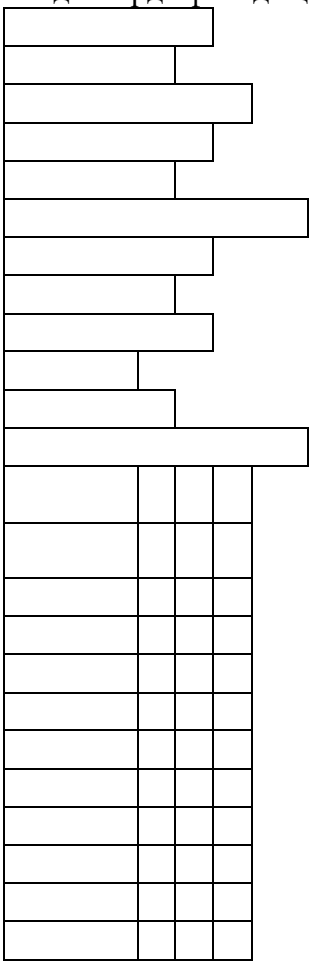
- Аралас файл.
- Сериялық файл.
- Индекс-сериялық файл.
- Индекстелген файл.
- Тікелей қатынау файлы (хэш).

Аралас файл

Файлды ұйымдастырудың ең аз күрделі формасын аралас (pile) деп атауға болады. Деректер қабылдау тәртібімен жинақталады. Әр жазба бір деректер пакетінен тұрады. Бұл

форма деректердің бүкіл массасын жинақтауды және оларды сақтауды жеңілдетеді. Жазбаларда әртүрлі өрістер де, әртүрлі ретпен орналастырылған өрістер де болуы мүмкін. Сондықтан әр өріс өзін-өзі сипаттауы керек, оның ішінде мағынасы да, аты да бар. Әр өрістің ұзындығы бөлгішті қолдану арқылы анық көрсетілуі керек немесе ішкі өріс ретінде нақты қосылуы керек.

Аралас файлдың құрылымы жоқ болғандықтан, жазбаға кіру барлық файл жазбаларын толығымен сұрыптау арқылы жүзеге асырылады. Егер белгілі бір өрісті қамтитын барлық жазбаларды немесе белгілі бір мәні бар өрісті қамтитын барлық жазбаларды табу қажет болса, онда іздеу бүкіл файл бойынша жүргізілуі керек. Көптеген жағдайларда файлдың бұл түрі жарамсыз.



Айнымалы ұзындық жазбалары

Тұрақты ұзындықтағы жазбалар

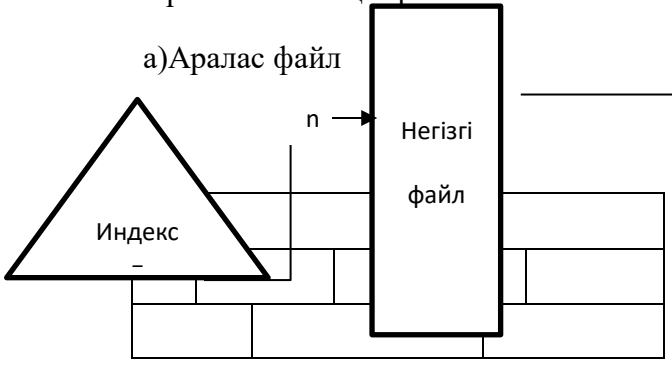
Өрістердің бекітілген жиынтығы және бекітілген тәртіп

Айнымалы өрістер жиынтығы

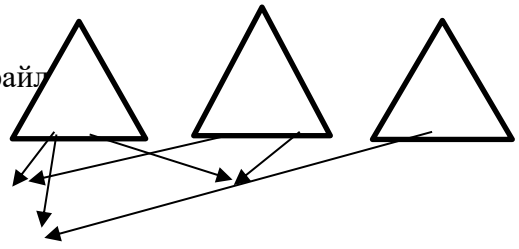
Хронологиялық тәртіп

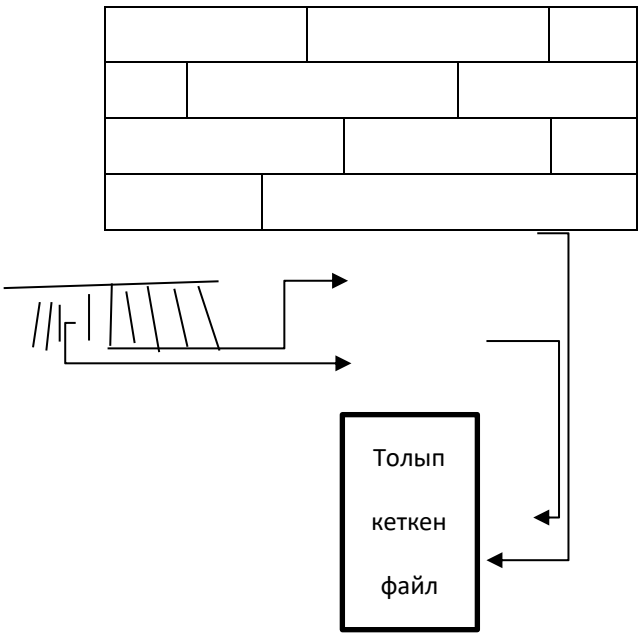
Кілттік өріске негізделген дәйекті тәртіп

а)Аралас файл



б) Дәйекті файл





в)индексті-дәйекті файл г) индекстелген файл

Сурет. 11.3. Файлдарды ұйымдастырудың негізгі тәсілдері

Сериялық файл

Файлдық құрылымның ең көп таралған түрі-сериялық файл. Онда жазбалар үшін бекітілген формат қолданылады. Барлық жазбалар бірдей ұзындыққа ие және белгілі бір ретпен ұйымдастырылған бірдей ұзындықтағы өрістерден тұрады. Әр өрістің ұзындығы мен орны белгілі болғандықтан, тек өріс мәндері сақталады; файл құрылымының атрибуттары-әр өрістің аты мен ұзындығы.

Бір нақты өріс, әдетте әр жазбадағы бірінші өріс, әр түрлі жазбалардың негізгі мәндері әрдайым әр түрлі болатындай етіп жазбаны ерекше түрде анықтайтын кілт өрісі деп аталады. Сонымен қатар, жазбалар "кілт" тізбегінде сақталады: мәтіндік кілт үшін алфавиттік ретпен және сандық кілт үшін сандық ретпен.

Индекс-сериялық файл

Индекс-сериялық файл сериялық файлдың негізгі ерекшелігін сақтайды: жазбалар негізгі өріс мәндеріне негізделген ретпен ұйымдастырылған. Бірақ сипатталған ұйыммен екі ерекшелік қосылды: еркін кіруді қолдау үшін файл индексі және толып кету файлы. Индекс қажетті жазбаны жылдам іздеуге мүмкіндік береді. Толып кету файлы сериялық кіру файлы пайдаланатын журнал файлына ұқсас, бірақ ол алдыңғы жазбаның көрсеткішіне сәйкес жазбалар орналастырылатын етіп ұйымдастырылған.

Индекстелген файл

Икемділікке қол жеткізу үшін іздеу объектісі бола алатын өрістің әр түрі үшін көбірек индекстерді қолдану қажет. Жалпыланған индекстелген файлда жазбаларға қол жеткізу тек олардың индекстері бойынша жүзеге асырылады. Нәтижесінде, кем дегенде бір индекстегі көрсеткіш осы жазбаға сілтеме жасағанға дейін жазбаларды орналастыруда ешқандай шектеулер жоқ. Сонымен қатар, мұндай файлда айнымалы ұзындықтағы жазбаларды орындау оңай.

Индекстердің екі түрі қолданылады. Толық индекс негізгі файлдың әр жазбасы үшін бір элементтен тұрады. Индекстің өзі іздеуді жеңілдету үшін сериялық файл ретінде ұйымдастырылған. Жеке индекс бізді қызықтыратын өріс бар жазбаларға арналған элементтерді қамтиды. Негізгі файлға жаңа жазба қосқан кезде барлық индекстік файлдарды жаңарту қажет. Индекстелген файлдарды, ең алдымен, ақпаратқа қол жеткізу

уақыты өте маңызды болып табылатын қосымшалар пайдаланады және файлдағы барлық жазбаларды өңдеу сирек қажет. Мысал ретінде әуе билеттеріне тапсырыс беру немесе түгендеу жүйелерін келтіруге болады.

Тікелей қатынау файлы

Тікелей қатынау файлы немесе хэш файлы файлдарды дискіде сақтау кезінде белгілі мекен-жайы бар блокқа тікелей кіру мүмкіндігін пайдаланады. Сериялық файлдардағы және индекстелген сериялық файлдардағы сияқты, әр жазбада кілт өрісі болуы керек. Алайда, деректерді дәйекті орналастыру тұжырымдамасы мұнда қолданылмайды.

Тікелей қатынау файлы негізгі мәндерді хэштеуді қолданады. Тікелей қатынау файлдары өте жылдам қол жетімділік қажет болған кезде, белгіленген ұзындықтағы жазбаларда, сондай-ақ барлық жазбаларға қол жеткізілген жағдайда қолданылады. Мысалдар каталогтар, бағалар тізімдері, кестелер және атаулар тізімдері болуы мүмкін.

11.3. Файл каталогтары

Файлдарды басқару жүйесі мен файлдар жиынтығы арасындағы байланыс файл каталогы болып табылады. Каталогта атрибуттар, орналасқан жер және тиесілі файлдар туралы ақпарат бар. Бұл ақпараттың көп бөлігі, әсіресе сақтауға байланысты ақпарат операциялық жүйенің бақылауында. Каталогтың өзі-бұл файлдарды басқарудың әртүрлі кіші бағдарламаларына қол жетімді файл . Каталогтағы кейбір ақпарат пайдаланушылар мен қосымшаларға қол жетімді болғанымен, бұл қол жетімділік жанама (жүйелік кіші бағдарламалар арқылы). Осылайша, пайдаланушылар каталогқа тікелей қол жеткізе алмайды (тіпті тек оқу режимінде).

Пайдаланушы тұрғысынан каталог пайдаланушылар мен қосымшаларға белгілі файлдардың аттарын файлдардың өзінде көрсетеді. Осылайша, каталогтағы әр жазбада Файл атауы болады. Іс жүзінде барлық жүйелер әртүрлі типтегі және ұйымдастыру тәсілдеріндегі файлдармен жұмыс істейді және бұл ақпаратты каталог ұсынады. Әр файл туралы ақпараттың маңызды категориясы-оны дискіде сақтауға байланысты ақпарат, оның ішінде файл өлшемі мен орналасқан жері. Бірлескен жүйелерде файлдарға қол жеткізуді басқару үшін ақпарат беру өте маңызды. Әдетте файлдың бір иесі болады, ал қалған пайдаланушыларға белгілі бір қол жетімділік құқығы берілуі мүмкін. Сонымен, файлды пайдалану туралы ақпарат ағымдағы файлды басқару үшін де, онымен жұмыс істеу тарихын жазу үшін де қажет.

Ақпаратты сақтау әдістері әр түрлі жүйелерде әр түрлі болады. Кейбір ақпаратты файлға қатысты тақырып жазбасында сақтауға болады. Бұл каталогтағы ақпараттың мөлшерін азайтады, нәтижесінде каталог негізгі жадқа толығымен жүктелуі мүмкін (және, тиісінше, онымен жұмыс істеу жылдамдығы күрт артады).

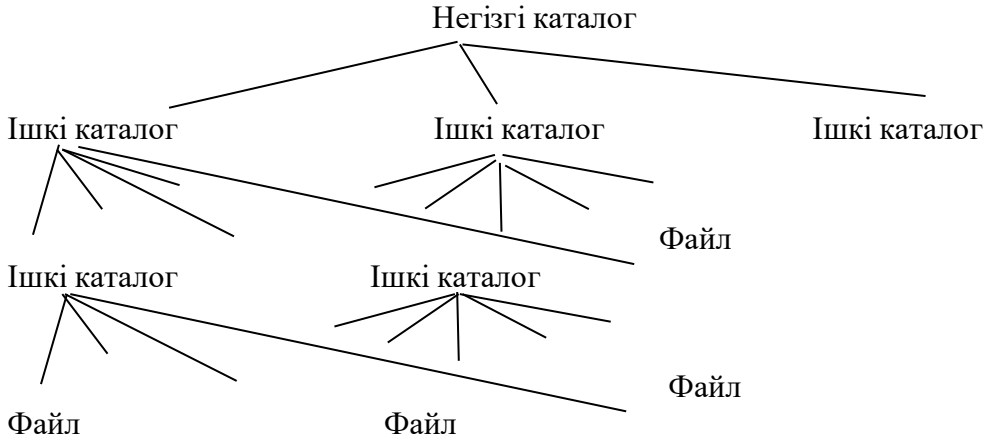
Каталогтың қарапайым құрылымы-әр файл үшін бір жазбалардың тізімі. Бұл құрылымды қарапайым сериялық файл ретінде ұсынуға болады, онда негізгі өріс Файл атауы болады. Бірақ бұл технология көп қолданушы жүйесіне де, тіпті көптеген файлдары бар жалғыз пайдаланушыларға да жарамайды.

Каталогтың үстінде жасалуы мүмкін операциялардың түрлерін қарастырамыз.

- **Іздеу.** Пайдаланушы немесе қолданба файлға кірген кезде каталогта сол файл туралы жазбаны іздеу қажет.
- **Файл жасау.** Жаңа файл жасаған кезде каталогқа сәйкес элементті қосу керек.
- **Файлды жою.** Файлды каталогтан жойған кезде тиісті элемент жойылуы керек.
- **Каталогтағы файлдар тізімін жасаңыз.** Жұмыс барысында каталогтың барлық мазмұнын (немесе оның бір бөлігін) сұрау қажет болуы мүмкін. Әдетте, пайдаланушының осы сұрауына берілген. Пайдаланушыға тиесілі барлық файлдардың тізімі, сондай-ақ әр файлдың кейбір атрибуттары қайтарылады (мысалы, оның түрі, қол жетімділік туралы ақпарат, пайдалану туралы ақпарат).
- **Каталогты жаңарту.** Кейбір файл атрибуттары каталогта сақталатындықтан, олардың кем дегенде біреуін өзгерту тиісті каталог элементіне өзгертулер енгізуді талап етеді.

Барлық осы операцияларды қолдау үшін қарапайым тізім қолайлы емес. Бірінші жуықтаудағы осы мәселелерді шешу екі деңгейлі схема болуы мүмкін, онда негізгі каталогтан басқа әр пайдаланушы үшін каталогтар бар. Негізгі каталогта әр пайдаланушы каталогына арналған жазбалар бар, оны мекен-жай және кіруді басқару туралы ақпаратпен қамтамасыз етеді. Әрбір пайдаланушы каталогы-бұл Пайдаланушының файлдарының қарапайым тізімі. Мұндай құрылым атаулардың тек бір пайдаланушының файлдар жиынтығында ерекше болуы керек және файлдық жүйе каталогқа кіруді шектеуді оңай қамтамасыз ете алады дегенді білдіреді. Алайда, бұл жүйе пайдаланушыға файлдар жиынтығын құрылымдауға әлі де көмектесе алмайды.

Неғұрлым қуатты және икемді тәсіл-иерархиялық немесе ағаш тәрізді құрылымды қолдану (11.4-сурет). Бұрынғыдай, негізгі каталог бар, оның ішінде пайдаланушы каталогтары орналасқан. Өз кезегінде, осы жеке каталогтардың әрқайсысында (кез-келген деңгейде) ішкі каталогтар мен файлдар болуы мүмкін.



Сурет. 11.4. Ағаш тәрізді каталог құрылымы

Атау

Пайдаланушыға файлға оның символдық атауы бойынша қол жеткізу мүмкіндігі қажет. Жүйедегі әр файлдың бірегей атауы болуы керек, сондықтан файлға жүгіну анық емес.

Ағаш каталогын пайдалану бірегей атауларды тағайындаудың күрделілігін азайтады. Жүйедегі кез-келген файлға түбірлік каталогтан ішкі каталогтардың тармақтары арқылы жетуге болады. Файл атауымен аяқталатын ішкі каталог атауларының сериясы файлдың толық атауын (жолын) анықтайды.

11.4. Файлдарды ортақ пайдалану

Көп қолданушы жүйесінде файлдарды пайдаланушылармен бөлісу мүмкіндігі әрдайым қажет. Бұл жағдайда екі сұрақ туындайды: қол жеткізу құқықтары және бір уақытта қол жеткізуді басқару.

Қол жеткізу құқықтары

Файлдық жүйе көптеген пайдаланушылар тарапынан файлға басқарылатын қол жетімділіктің икемді мүмкіндігін қамтамасыз етуі керек. Ол белгілі бір файлға қол жеткізуді басқарудың әртүрлі нұсқаларын ұсынуы керек. Әдетте файлға кіру құқығы пайдаланушыларға немесе пайдаланушылар тобына беріледі. Қол жеткізу құқықтарының кең ауқымы қолданылады. Бұл тізімде кейбір файлға қатысты белгілі бір пайдаланушыға берілуі мүмкін қол жетімділік құқықтары көрсетілген.

- **Болмау.** Пайдаланушы файлдың бар-жоғын анықтай алмайды, оған қол жеткізуді айтпағанда. Мұндай шектеуді қамтамасыз ету үшін пайдаланушыға осы файлдағы пайдаланушы каталогын оқуға тыйым салу керек.

- **Білім.** Пайдаланушы файлдың бар-жоғын анықтап, оның иесін орната алады. Осыдан кейін пайдаланушы файлға қосымша кіру құқығын алу үшін иесіне жүгіне алады.
- **Орындау**. Пайдаланушы бағдарламаны жүктей және орындай алады, бірақ файлды көшіре алмайды. Пайдаланушы бағдарламалары көбінесе осындай шектеулермен қол жетімді.
- **Оқу.** Пайдаланушы файлды кез-келген мақсатта оқи алады, соның ішінде көшіру және орындау. Кейбір жүйелер оқу мен көшіруді ажырата алады. Бұл жағдайда файлдың мазмұнын дисплейге шығаруға болады, бірақ пайдаланушы файлды көшіре алмайды.
- **Қосу.** Пайдаланушы файлға деректерді көбінесе оның соңында ғана қоса алады, бірақ ол файл мазмұнын өзгерте немесе жоя алмайды. Бұл құқық әртүрлі көздерден деректерді жинау кезінде пайдалы.
- **Жаңарту.** Пайдаланушы файлға деректерді өзгерте, жоя немесе қоса алады. Әдетте бұл файлға бастапқы жазу, толық немесе ішінара қайта жазу, деректерді толық немесе ішінара жою операцияларын қамтиды. Кейбір жүйелер жаңартудың әртүрлі деңгейлерін ажыратады.
- **Қорғауды өзгерту.** Пайдаланушы басқа пайдаланушыларға берілген кіру құқықтарын өзгерте алады. Әдетте бұл құқық тек файл иесіне тиесілі. Кейбір жүйелерде пайдаланушы бұл құқықты басқа пайдаланушыларға тарата алады. Бұл механизмді теріс пайдаланудан қорғау үшін файл иесі қандай құқықтарды өзгертуге болатындығын анықтай алады.
- **Жою.** Пайдаланушы файлды файлдық жүйеден жоя алады. Бұл құқықтарды белгілі бір иерархия ретінде қарастыруға болады, онда бір қол жеткізу құқығына ие болу иерархияда оған дейінгі барлық құқықтарға ие болады.

Пайдаланушылардың бірі файл иесі ретінде анықталады; бұл әдетте файлды жасаған пайдаланушы. Иесі аталған барлық құқықтарға ие және басқа пайдаланушыларға құқықтар бере алады. Әр түрлі пайдаланушы сыныптарына қол жеткізуге болады.

- **Нақты пайдаланушы**. Жеке пайдаланушылар өздерінің идентификаторлары арқылы анықталады.
- **Пайдаланушылар тобы**. Жеке анықталмаған көптеген пайдаланушылар. Жүйеде пайдаланушылардың топқа мүшелігін бақылаудың кейбір әдісі болуы керек.
- **Барлығы.** Жүйеге рұқсаты бар барлық пайдаланушылар. Ортақ пайдалану файлдары барлық пайдаланушылар үшін қол жетімді.

Бір уақытта қол жеткізу

Егер кіру рұқсаттары файлды бірнеше пайдаланушыға қосуға немесе жаңартуға мүмкіндік берсе, онда бұл жағдайда операциялық жүйе немесе файлдарды басқару жүйесі пайдаланушылардың файлмен жұмыс жасауының белгілі бір тәртібін ұйымдастыруы керек. Ортақ қол жетімділікті жобалау кезеңінде өзара алып тастау және оқшаулау мәселелерін шешу қажет.

11.5. Екінші жадты басқару

Сыртқы жад құрылғысында сақталған файл блоктар жиынтығынан тұрады. Операциялық жүйе немесе файлдарды басқару жүйесі файлдарға блоктарды бөлуге жауап береді, бұл басқаруға байланысты екі мәселені тудырады. Бірінші мәселе-екінші жад кеңістігі файлдар арасында бөлінуі керек, ал екіншісі – бөлуге болатын кеңістікті бақылау қажет. Бұл міндеттер бір-бірімен тығыз байланысты екені айтпаса да түсінікті. Басқаша айтқанда, файлды орналастыру кезінде қолданылатын әдіс бос орынды басқару әдісіне әсер етуі мүмкін.

Файлдарды орналастыру

Файлдарды орналастыру кезінде бірнеше сұрақтар туындайды.

1. Файлды жасаған кезде бірден максималды кеңістікті бөлу керек пе?

2. Файлға бір немесе бірнеше үздіксіз бірліктер немесе бөліктер түрінде орын беріледі. Мұндай бөліктің мөлшері бір блоктан бүкіл файлға дейін өзгеруі мүмкін. Файлды орналастыру кезінде қандай бөлікті пайдалану керек?

3. Файл бөліктерін есепке алу үшін деректер құрылымының немесе кестенің қандай түрі қолданылады? Мұндай кесте әдетте файлдарды орналастыру кестесі деп аталады (file allocation table – FAT) және DOS және басқа да операциялық жүйелерде бар.

Алдын-ала және динамикалық орналастыру

Файлды жасауды сұраған кезде оны алдын-ала орналастыру стратегиясы файлдың максималды өлшемін алдын-ала білуді талап етеді. Бағдарламаларды құрастыру, жиынтық деректер файлдарын алу немесе файлды басқа жүйеден желі арқылы жіберу сияқты кейбір жағдайларда бұл мән сенімді түрде бағалануы мүмкін. Алайда, көптеген қосымшалар үшін мұндай бағалау қиын, тіпті мүмкін емес. Мұндай жағдайларда қолданбалы бағдарламаларды пайдаланушылар мен әзірлеушілер тапсырыс берілген өлшем жеткіліксіз болған жағдайда болмас үшін бағаланатын файл өлшемін асырып жіберуі керек. Мұндай стратегия дискілік кеңістікті шамадан тыс тұтынуға әкелетіні анық, және қажет болған жағдайда файлдардың бөліктері үшін орын бөлінетін динамикалық орналастыруды қолданған жөн.

Қызмет ету мөлшері

Тағы бір сұрақ файлға бөлінген бөліктің мөлшеріне қатысты. Бір шекті жағдай-бүкіл файлды орналастыру үшін жеткілікті үлкен бөлікті бөлу. Тағы бір шекті жағдай – дискідегі кеңістікті бөлу бір блокта жүреді. Қызмет көрсету мөлшерін таңдағанда, бір файлдың тиімділігі мен жүйенің жалпы тиімділігі арасындаромаға келу қажет. Баламалы нұсқаларды талдау барысында келесі ойлар қарастырылады [268].

1. Кеңістіктің үздіксіздігі өнімділікті арттырады, атап айтқанда келесі түрдегі операцияларды орындау кезінде, әсіресе транзакцияға бағытталған операциялық жүйеде орындалатын транзакциялар.
2. Кішкентай бөліктердің көп болуы орналастыру туралы ақпаратты басқаруға қажетті кестелердің көлемін арттырады.
3. Белгіленген мөлшердегі бөліктердің болуы (мысалы, блоктар) кеңістікті қайта бөлуді жеңілдетеді.
4. Айнымалы ұзындықтың немесе кішігірім мөлшердің болуы артық үлестірудің салдарынан пайдаланылмаған кеңістіктің жоғалуын азайтады.

Әрине, бұл жағдайлар бір-бірімен байланысты және оларды бірге қарау керек. Нәтижесінде біз екі негізгі нұсқаны аламыз.

1. Айнымалы ұзындықтың үлкен үздіксіз бөліктері. Жақсы өнімділікті қамтамасыз етіңіз. Айнымалы Өлшем пайдаланылмаған жерлерді болдырмауға көмектеседі, ал файлдарды орналастыру кестелері кішкентай. Алайда, дискілік кеңістікті қайта пайдалануды ұйымдастыру қиын.
2. Блоктар. Бекітілген мөлшердегі кішкене бөліктер үлкен иілуді қамтамасыз етеді. Бұл жағдайда үлкенірек немесе күрделі құрылымы бар кестелер қажет болуы мүмкін. Үздіксіздік жоқ, ал файлға арналған блоктар қажет болған жағдайда ерекшеленеді.

Кез-келген нұсқа алдын-ала және динамикалық орналастырумен үйлесімді. Бірінші опцияны қолданған кезде файл блоктардың бір үздіксіз тобы ретінде алдын-ала орналастырылады. Файлдарды орналастыру кестесінің қажеттілігі жоғалады. Тек бірінші блокты және орналастырылған блоктардың санын көрсету керек. Блок құрылымын таңдағанда, барлық қажетті бөліктер бір уақытта орналастырылады. Бұл fat файлының бекітілген өлшемі бар екенін білдіреді.

Тарату стратегиялары келесідей болуы мүмкін.

1. Біріншісі қолайлы. Бос блоктар тізімінен дұрыс өлшемдегі бірінші пайдаланылмаған үздіксіз блок-кілем тобын таңдау.
2. Ең жақсы қолайлы. Көлемі жеткілікті болатын ең аз пайдаланылмаған топты таңдау.

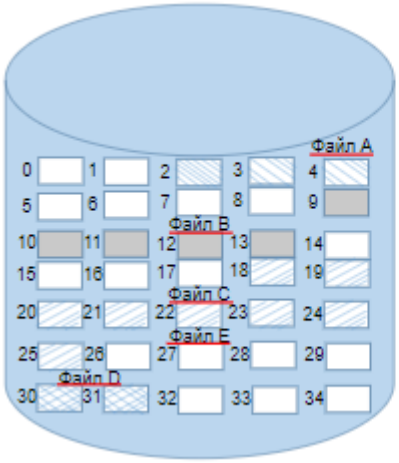
3. Ең жақын қолайлы. Локализацияны ұлғайту үшін соңғы орналастырылған файлға ең жақын орналасқан дұрыс өлшемдегі пайдаланылмаған топты таңдау.

Бұл стратегияларды модельдеу кезінде - файл түрлері, оларға қол жеткізу схемалары, көп функциялы дәреже, дискіні кэштеу, дискіні жоспарлау және т. б. сияқты көптеген факторларды ескеру қажет.

Файлдарды орналастыру әдістері

Файлдарды орналастырудың ең көп қолданылатын үш әдісі-үздіксіз, тізбекті және индекстелген.

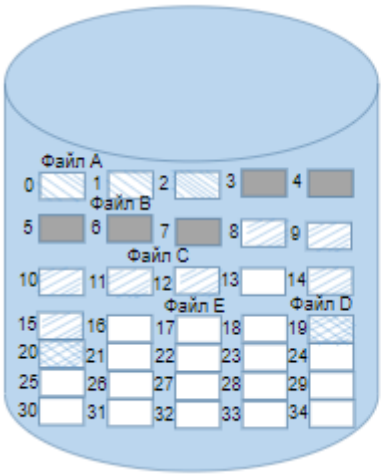
Жасалған файлды үздіксіз орналастырған кезде блоктардың жеке үздіксіз жиынтығы бөлінеді (11.5-сурет). Осылайша, бұл өзгермелі мөлшердегі бөліктері бар алдын-ала орналастыру стратегиясы. Әр файл үшін орналастыру кестесіне файлдың бастапқы блогы мен ұзындығын анықтайтын бір ғана элемент қажет. Үздіксіз орналастыру жеке сериялық қол жетімділік файлы тұрғысынан ең жақсы болып табылады. Енгізу-шығару өнімділігін арттыру үшін бір уақытта көптеген блоктарды бір уақытта өңдеуге болады. Алайда, файлды үздіксіз орналастыру кейбір мәселелерді тудырады. Сонымен, сыртқы фрагментация пайда болады, бұл тиісті мөлшердегі үздіксіз блоктарды іздеуді қиындатады. Кейде дискідегі қажетті кеңістікті босату үшін қаптаманы орындау қажеттілігі туындайды (11.6-сурет).



Файлдарды орналастыру кестесі

Файл аты	Бастапқы блок	Ұзындығы
Файл A	2	3
Файл B	9	5
Файл C	18	8
Файл D	30	2
Файл E	26	3

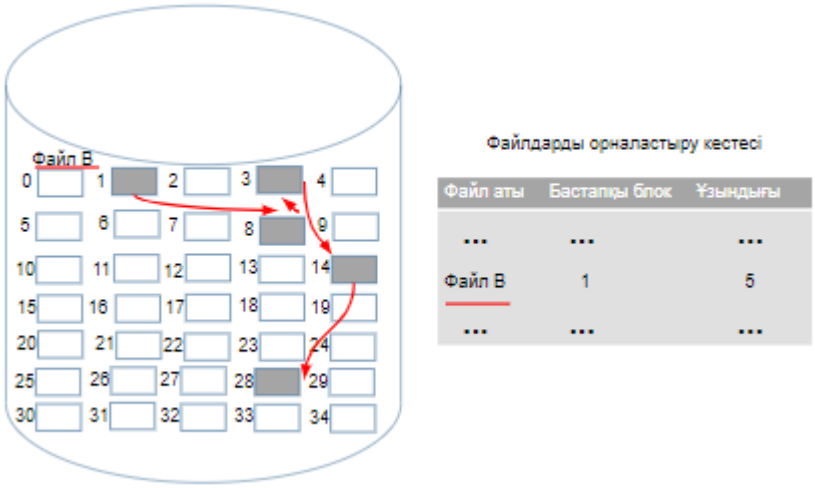
Сурет. 11.5. Файлдарды үздіксіз орналастыру(Көшірілген)



Файлдарды орналастыру кестесі

Файл аты	Бастапқы блок	Ұзындығы
Файл A	0	3
Файл B	3	5
Файл C	8	8
Файл D	19	2
Файл E	16	3

Сурет. 11.6. Файлдарды үздіксіз орналастыру (тығыздалғаннан кейін) (Көшірілген)

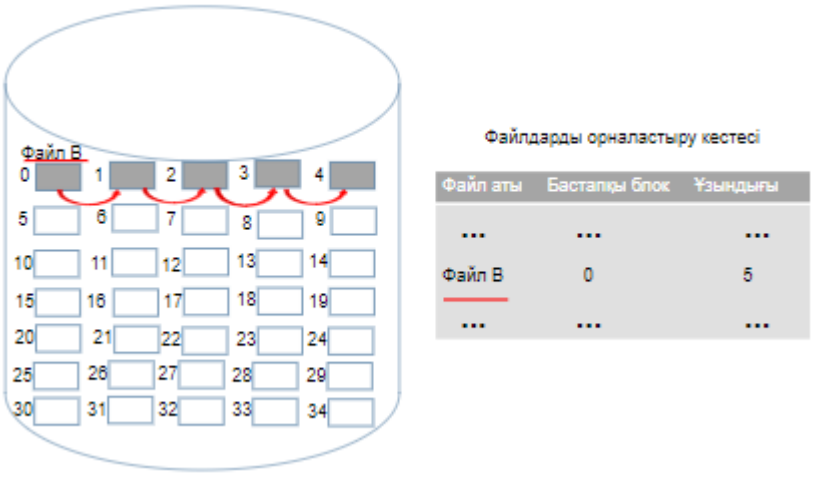


Сурет. 11.7. Файлдарды тізбектей орналастыру(Көшірілген)

Үздіксіз орналастыруға қарама-қарсы ретінде файлды тізбектей орналастыру әдісі жасалды (11.7-сурет). Әдетте орналастыру бір блоктан тұрады. Әр блокта тізбектегі келесі блокқа көрсеткіш бар. Бұл файлды орналастыру кестесінде бастапқы блок пен файлдың ұзындығын көрсететін әр файл үшін тек бір элемент қажет. Бұл схемада блоктарды алдын-ала бөлуді қолдануға болатындығына қарамастан, қажет болған жағдайда файлға блоктарды бөлу оңайырақ. Физикалық ұйымның бұл түрі дәйекті қол жеткізу файлдары үшін ең қолайлы, онда деректерді өңдеу дәйекті түрде жүзеге асырылады.

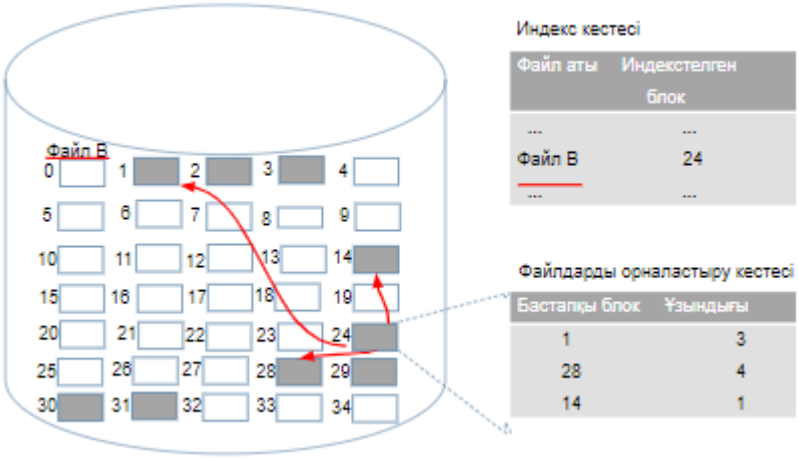
бір уақытта бірнеше файл блоктарын жүктеу қажет болса, дискінің әр түрлі бөліктеріне бірқатар қоңыраулар қажет. Бұл мәселені шешу үшін кейбір жүйелер мезгіл-мезгіл файлдарды тығыздайды (сурет 11.8).

Индекстелген орналастыру файлдарды үздіксіз және тізбекті орналастырудың бірқатар мәселелерін шешеді. Бұл жағдайда файлдарды орналастыру кестесінде әр файл үшін бөлек бір деңгейлі индекс бар. Индекс файлдың әр бөлігі үшін бір жазбадан тұрады. Әдетте файл индекстері файлдарды орналастыру кестесінің бөлігі ретінде физикалық түрде сақталмайды; оның орнына файл индекстері бөлек блокта сақталады және файлды орналастыру кестесінің элементінде сол блок үшін көрсеткіш бар. Орналастыру бекітілген өлшемді блоктарға (11.9-сурет) және айнымалы өлшемді блоктарға (11.10-сурет) негізделуі мүмкін**.**



Сурет. 11.8. Файлдарды тізбекті орналастыру (тығыздалғаннан кейін) (Көшірілген)

Сурет. 11.9. Блоктардың бөліктерімен индекстелген орналастыру(Көшірілген)



Сурет. 11.10. Айнымалы мөлшердегі бөліктері бар индекстелген орналастыру(Көшірілген)

Бос кеңістікті басқару

Таңдалған файлдарға кеңістікті басқару қажет болғандықтан, қазіргі уақытта ешқандай файлға бөлінбейтін кеңістік басқарылуы керек. Жоғарыда сипатталған файлдарды орналастыру технологиясының кез-келгенін қолдана отырып, дискідегі қандай блоктар бар екенін білуіңіз керек. Ол үшін файлдарды орналастыру кестесінен басқа, дискіні орналастыру кестесі қажет. Оның ықтимал іске асырылуын қарастыру қажет.

Биттік кестелер

Бұл әдіс әр диск блогы үшін бір биттен тұратын векторды қолданады. Әрбір нөл бос блокқа сәйкес келеді, ал әрбір бірлік пайдаланылған блокқа сәйкес келеді. Биттік кестенің артықшылықтарының бірі-бір бос блокты немесе үздіксіз бос блоктар тобын табудың салыстырмалы қарапайымдылығы. Екінші артықшылығы-оның барлық мүмкін көріністерінің ішіндегі ең кішісі. Дегенмен, дискіні орналастыру кестесі әсерлі өлшем болуы мүмкін. Тиісінше, биттік кестелерді қолданатын файлдық жүйелердің көпшілігі биттік кестенің ішкі диапазондарының мазмұнын қорытындылайтын қосалқы деректер құрылымдарын қолдайды. Мысалы, кестені логикалық түрде бірдей өлшемдегі бірнеше ішкі жолақтарға бөлуге болады. Жиынтық кестеде әрқайсысы үшін бос блоктар саны және ең үлкен үздіксіз блок туралы ақпарат болуы мүмкін. Егер бірнеше дәйекті блоктар қажет болса, файлдық жүйе сәйкес ішкі диапазонды табу үшін жиынтық кестені сканерлеп, содан кейін нақты іздеу жүргізе алады.

Бос бөліктер тізбегі

Бос бөліктерді әрбір бос бөліктің көрсеткіштері мен ұзындық мәндерін пайдаланып тізбектерге қосуға болады. Бұл әдіс кішігірім үстеме шығындарға әкеледі, өйткені дискіні

орналастыру кестесінің қажеті жоқ, тек тізбектің басталуын және бірінші бөліктің ұзындығын көрсету қажет. Бұл әдіс барлық файлдарды орналастыру әдістеріне сәйкес келеді. Егер орналастыру блок бойынша жүрсе, онда тізбектің жоғарғы жағындағы бос блокты таңдау және көрсеткіштің тізбектің басына және ұзындық мәніне сәйкес өзгеруі жүзеге асырылады. Айнымалы ұзындықтағы бөліктерді орналастыру кезінде бірінші қолайлы алгоритмді қолдануға болады-тізбектің бірінен біз үшін қолайлы деп тапқанға дейін бөлік тақырыптарын шығарып алу. Көрсеткіш пен ұзындық мәндері сәйкесінше реттеледі.

Сипатталған әдіс кемшіліктерге ие. Біраз уақыт қолданғаннан кейін диск қатты бөлшектенеді және көптеген бөліктер бір блокпен салыстырылатын ұзындыққа ие болуы мүмкін.

Индекстеу

Индекстеу кезінде бос орын файл ретінде қарастырылады (файлдарды орналастыру сияқты, мұнда индекс кестесі қолданылады). Тиімді жұмыс істеу үшін индекстер блоктарға емес, айнымалы мөлшердегі бөліктерге негізделуі керек. Дискідегі әрбір бос бөлік үшін кестедегі бір элемент сәйкес келеді. Бұл тәсіл файлдарды орналастырудың барлық әдістеріне тиімді қолдау көрсетеді.

Бос блоктардың тізімі

Бұл әдісте әр блокқа реттік нөмір беріледі, ал барлық бос блоктардың нөмірлерінің тізімі дисктің сақталған аймағында болады. Дискінің көлеміне байланысты бір блок нөмірін сақтау үшін 24 немесе 32 бит қажет. Сондықтан, бос блок тізімінің өлшемі сәйкес бит кестесінен 24 немесе 32 есе үлкен, сондықтан оны негізгі жадта емес, дискіде сақтау керек. Осыған қарамастан, ұсынылған әдіс өте жақсы. Келесі ережелерді қарастырыңыз.

1. Бос блоктар тізіміне бөлінген диск кеңістігі жалпы диск кеңістігінің 1% - дан азын алады. Блоктардың 32 биттік нөмірленуін пайдаланған кезде, әр 512 байт блок үшін тұтыну 4 байтты құрайды.
2. Дегенмен бос блоктар тізімі Негізгі жадта сақтау үшін тым үлкен, тізімнің кішкене бөлігін негізгі жадта сақтаудың екі тиімді әдісі бар: тізімді стек ретінде қарастыруға болатын кезде және тізім кезек түрінде ұйымдастырылған кезде.

Сипатталған стратегиялардың кез-келгенінде фондық ағынды қолдануға болады, ол үздіксіз бөлуді жеңілдету үшін тізімді негізгі жадта сұрыптайды.

Томдар

Том (көлем) - операциялық жүйе немесе қолданба деректерді сақтау үшін пайдалана алатын екінші жадтағы мақсатты секторлардың жиынтығы. Көлемдегі секторлар физикалық сақтау құрылғысында дәйекті болуы міндетті емес; оның орнына олар тек операциялық жүйеге немесе қолданбаға ұқсас болуы керек. Том кішігірім көлемдерді жинау мен біріктірудің нәтижесі болуы мүмкін. Қарапайым жағдайда бір диск бір дискіге тең. Көбінесе диск бөлімдерге бөлінеді, әр бөлім жеке көлем ретінде жұмыс істейді. Сондай-ақ, көбінесе бірнеше дискілер немесе бірнеше дискілердегі бөлімдер бір том ретінде қарастырылады.

Сенімділік

Келесі сценарийді қарастырайық.

1. А пайдаланушысы бар файлға ақпарат қосу үшін орналастыру сұрауын жасайды.
2. Сұраныс қабылданады және дискіде емес, негізгі жадта дискілік және файлдарды орналастыру кестелері жаңартылады.
3. Жүйеде ақау пайда болады, содан кейін қайта жүктеледі.
4. Б пайдаланушысы файлды орналастыру туралы сұрау салады және а файлы пайдаланушысының өтініші бойынша соңғы орналастырумен қабаттасатын кеңістікте орналасады.

5. А пайдаланушысы а пайдаланушының файлында сақталған сілтемені пайдаланып, оған бұрын бөлінген (және қазір Б пайдаланушысы алып жатқан) бөлікке қол жеткізеді.

Мәселе тиімділікті арттыру үшін жүйеде негізгі жадта дискіні орналастыру кестесінің көшірмесі болғандықтан туындайды. Мұндай қателіктердің алдын алу үшін файлды орналастыруды сұраған кезде бірқатар әрекеттерді орындау керек.

1. Дискідегі орналастыру кестесін бұғаттаңыз. Бұл сұранысты өңдеу аяқталғанға дейін басқа пайдаланушылардың кестеге өзгерістер енгізуіне жол бермейді.
2. Дискіні орналастыру кестесінде қол жетімді кеңістікті іздеңіз. Дискіні орналастыру кестесінің көшірмесі әрқашан негізгі жадта болады деп болжанады. Егер олай болмаса, алдымен кестені негізгі жадқа оқу керек.
3. Кеңістікті таратыңыз, жад пен дискідегі дискіні орналастыру кестесін жаңартыңыз. Дискіні орналастыру кезінде дискідегі көрсеткіштерді де жаңарту керек.
4. Файлдарды негізгі жадқа және дискіге орналастыру кестесін жаңартыңыз.
5. Дискіні орналастыру кестесін блоктан шығарыңыз.

Бұл әдіс қателіктерден қорғайды, бірақ, кішкене бөліктердің жиі таралуы өнімділіктің айтарлықтай төмендеуіне әкеледі. Бұл әсерді азайту үшін пакеттік орналастыру сызбасын қолдануға болады. Бұл жағдайда дискідегі бос бөліктер пакеті орналастыру үшін бөлінеді. Дискідегі тиісті бөліктер пайдаланылған деп белгіленеді. Пакеттік орналастыру негізгі жадта болуы мүмкін. Пакет таусылған кезде дискідегі файлдарды орналастыру кестесі жаңартылып, жаңа пакет таңдалады. Жүйе істен шыққан кезде пайдаланылған деп белгіленген дискінің бөліктері оларды қайта орналастырмас бұрын тазалануы керек.

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