

ЛЕКЦИЯ № 4. Жадты басқару

Жад нақты басқаруды қажет ететін өте маңызды ресурс болып табылады. Бұл күндері үйдегі орташа компьютердің жады 1960 жылдардың басында әлемдегі ең үлкен компьютер болған IBM 7094 ресурстарынан он мың есе көп болса да, компьютерлік бағдарламалардың көлемі жад көлеміне қарағанда тез өсуде. Паркинсон Заңын келесідей парафразалауға болады: "Бағдарламалар оларды орналастыруға болатын барлық жадты толтыруға тырысып, көлемі ұлғаяды".

Ең дұрысы, әрбір бағдарламашы тек көлемі мен жұмыс жылдамдығы бойынша шексіз жадқа ие болғысы келеді, ол сонымен қатар қуат өшірілген кезде мазмұнын жоғалтпайды. Біз осылай белгілегендіктен, неге жадты өте арзан етпеске? Өкінішке орай, қолданыстағы технологиялар бізге әлі қалағанын бере алмайды. Мүмкін сіз осындай жадты құрудың жолын ойлап таба аласыз.

Сонда біз немен қанағаттануымыз керек? Уақыт өте келе жад иерархиясының тұжырымдамасы жасалды, оған сәйкес компьютерлерде бірнеше мегабайт өте жылдам, қымбат және өзгермелі кэш жады, бірнеше гигабайт жады, жылдамдығы да, бағасы да орташа, сонымен қатар бірнеше терабайт жады бар. өте баяу, салыстырмалы түрде арзан диск жетектерінде, мысалы, ауыстырылатын дискілерді айтпағанда DVD және USB флэш құрылғылары сияқты. Бұл иерархияны абстракцияға, яғни ыңғайлы модельге айналдыру, содан кейін сол абстракцияны басқару – және амалдық жүйенің міндеті бар.

Жад иерархиясын (немесе оның бір бөлігін) басқаратын операциялық жүйенің бөлігі менеджер немесе жад менеджері деп аталады. Ол жадты тиімді басқаруға арналған және жадтың қандай бөліктері қолданылатынын қадағалап, оған қажет процестерге жадты бөліп, процестер аяқталған кезде жадты босатуы керек.

Абстракцияларды қолданбай жад

Жадтың қарапайым абстракциясы кез-келген абстракцияның толық болмауы деп санауға болады. Ерте әмбебап машиналар (1960 жылға дейін), ерте шағын компьютерлер (1970 жылға дейін) және ерте дербес компьютерлер (1980 жылға дейін) жад абстракциясын қолданбаған. Әр бағдарлама физикалық жадты көрді. Бағдарлама пәрменді орындаған кезде

MOV REGISTER1,1000

компьютер 1000 физикалық жад ұяшығының мазмұнын REGISTER1-ге ауыстырды. Осылайша, бағдарламашыға берілген жад моделі қарапайым физикалық жады болды, адрестер жиынтығы 0-ден белгілі бір максималды мәнге дейін, мұнда әр адрес қандай да бір бит саны бар ұяшыққа сәйкес келді (әдетте 8).

Мұндай жағдайларда бірден екі жұмыс істейтін бағдарламаның жадындағы мазмұн мүмкін болмады. Егер бірінші бағдарлама, мысалы, 2000 ұяшығына жаңа мәнді жазса, онда ол екінші бағдарлама сақтаған мәнді өшіреді. Жұмыс мүмкін болмады және екі бағдарлама да бірден сәтсіздікке ұшырады.

Физикалық жадтың өзі жад моделі ретінде әрекет ететін жағдайларда да жадты пайдаланудың бірнеше нұсқалары мүмкін. Олардың үшеуі суретте көрсетілген. 4.1. Операциялық жүйе мүмкін (сурет. 4.1, а) мекенжайлардың төменгі жағында, жедел жад құрылғысында (ЖЖҚ) немесе басқаша түрде кездейсоқ қол жетімді жадта-RAM (RandomAccessMemory) орналастырылады. Ол сондай-ақ тұрақты сақтау құрылғысында (ROM) немесе ROM-да (read-OnlyMemory), жоғарғы жад мекенжайларында (сурет. 4.1, б). Немесе құрылғы драйверлері жадтың жоғарғы адрестерінде, ROM-да, ал жүйенің қалған бөлігі жедел жадта, ең төменгі жағында болуы мүмкін (сурет. 4.1, в). Бірінші модель бұрын әмбебап машиналарда және шағын компьютерлерде қолданылған, ал басқа машиналарда сирек кездеседі. Екінші модель кейбір PDA және ендірілген жүйелерде қолданылды. Үшінші модель ерте дербес компьютерлерде қолданылды (мысалы, MS-DOS жұмыс істейтіндер), мұнда жүйенің ROM-да орналасқан бөлігі негізгі Енгізу – шығару жүйесі-BIOS (BasicInputOutputSystem) деп аталды. Суретте көрсетілген модельдердің жетіспеушілігі. 4.1 және в, бұл пайдаланушы бағдарламасындағы қате операциялық жүйені өшіруі мүмкін және, мүмкін, өте зиянды салдары бар.

Жад абстракциясы жоқ жүйеде кем дегенде параллельдікке қол жеткізудің жалғыз жолы-бағдарламаны бірнеше ағынмен іске қосу. Процестегі барлық ағындар бірдей жад бейнесін көреді деп болжанғандықтан, оларды мәжбүрлеп қолдану проблема тудырмайды. Бұл идея өте тиімді болғанымен,

ол кең қолданыс таппады, өйткені адамдар көбінесе бір-бірімен байланысты емес бағдарламаларды бір уақытта іске қосуды қажет етеді және бұл ағындардың абстракциясы қамтамасыз етпейді. Сонымен қатар, жадтың абстракциясын қамтамасыз етпейтін мұндай қарабайыр жүйелер ағындардың абстракциясын қамтамасыз ете алмайды.

Дегенмен, жад абстракциялары болмаса да, бір уақытта бірнеше бағдарламаны іске қосу әбден мүмкін. Ол үшін амалдық жүйе жадтың барлық мазмұнын дискідегі файлға сақтап, содан кейін келесі бағдарламаны жүктеп, іске қосуы керек. Жадта бір уақытта бір ғана бағдарлама болғандықтан, қақтығыстар болмайды. Бұл тұжырымдама деректерді ауыстыру (немесе своп) деп аталады және сәл кейінірек қарастырылады.

Мұндай жад жүйесінің құрылғысымен процестер, әдетте, бір-бірден басталуы мүмкін. Пайдаланушы пәрменді тергеннен кейін амалдық жүйе сұралған бағдарламаны дискіден жадқа көшіреді, содан кейін ол осы бағдарламаны орындайды. Процесс аяқталғаннан кейін амалдық жүйе шақыру белгісін көрсетеді және жаңа команданың енгізілуін күтеді. Пәрменді алғаннан кейін ол жадқа жаңа бағдарламаны жүктейді, оны бірінші бағдарламаның үстіне жазады



Сурет 4.1. Операциялық жүйе мен бір пайдаланушы процесі болған кезде жадты ұйымдастырудың үш қарапайым әдісі (басқа нұсқалар бар)

Кейбір арнайы қосымша жабдықтар болған кезде, свопты қолданбай-ақ бірнеше бағдарламаны параллель іске қосу мүмкіндігі пайда болады. IBM 360-тың алғашқы модельдерінде бұл мәселе келесідей шешілді: Жад 2 Кб блоктарға бөлінді, олардың әрқайсысына орталық процессордан тыс арнайы регистрлерде қамтылған 4 биттік қауіпсіздік кілті берілді. 1 Мбайт жады бар машинада осындай 4 биттік регистрлердің тек 512-сі болуы керек еді, ал барлық кілттер қоймасы 256 байт жадты алды. Бағдарламаның күй сөзі (program Status Word (PSW)) 4 биттік кілтті де қамтыды. IBM 360 аппараттық құралы іске қосылған процестің PSW кілтінен өзгеше қорғаныс кілтімен жадқа қол жеткізу әрекетін тоқтатты. Қорғаныс кілттерін тек Операциялық жүйе өзгерте алатындықтан, пайдаланушы процестері бір-біріне және операциялық жүйенің жұмысына араласудан қорғалған.

Негізгі әдебиеттер: [1,2,3].

Қосымша әдебиеттер: [8,11].

Бақылау сұрақтары:

1. ОЖ-де жад менеджері деп не аталады?
2. MOV REGISTER 1.1000 пәрменін түсіндіріңіз.
3. "Кездейсоқ қол жетімді жад" терминін қалай түсінесіз?
4. ОЖ және бір пайдаланушы процесі болған кезде жадты ұйымдастырудың үш қарапайым әдісін сипаттаңыз.
5. Негізгі Енгізу/шығару жүйесі (BIOS) дегеніміз не?