

Мультипроцессор

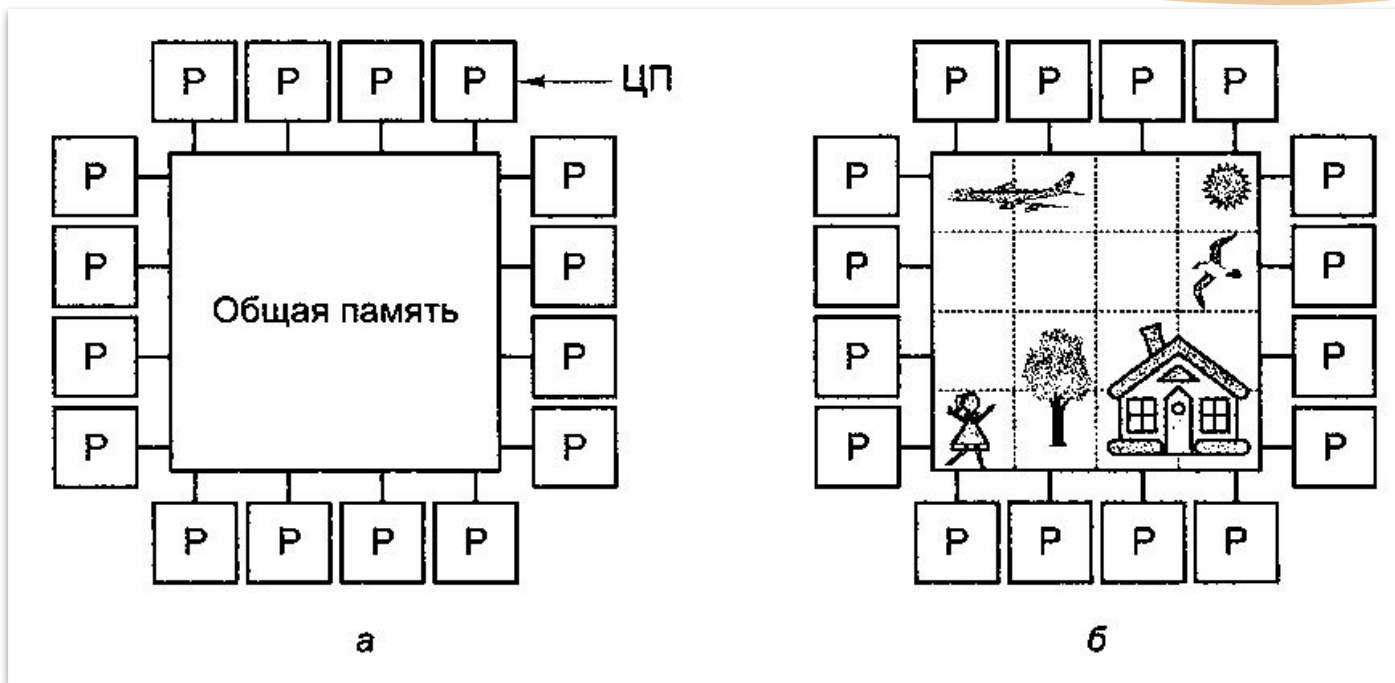
Ақпарат алмасу

Кез келген параллельді компьютерлік жүйеде бір тапсырманың әртүрлі бөліктерін орындайтын процессорлар ақпарат алмасу үшін бір-бірімен қандай да бір түрде әрекеттесуі керек. Айырбастау нақты қалай болуы керек? Ол үшін екі стратегия ұсынылып, іске асырылды: мультипроцессорлар және мультикомпьютерлер. Стратегиялар арасындағы негізгі айырмашылық - ортақ жадтың болуы немесе болмауы. Бұл айырмашылық мұндай жүйелердің дизайнына, дизайнына және бағдарламалануына, сондай-ақ олардың құны мен өлшеміне әсер етеді.

Мультипроцессорлар

Барлық процессорлары ортақ физикалық жадты ортақ пайдаланатын параллельді компьютерді мультипроцессорлық немесе ортақ жады жүйесі деп атайды. Көп процессорда бірге жұмыс істейтін барлық процестер ортақ жадқа салыстырылған бір виртуалды мекенжай кеңістігіне ие болуы мүмкін. Жүктеп алу және жүктеу командаларын қолданатын кез келген процесс жадтан сөзді оқи алады немесе сөзді жадыға жаза алады. Басқа ештеңе талап етілмейді. Екі процесс ақпаратты оңай алмасу мүмкіндігіне ие - бұл үшін олардың біреуі жай ғана ортақ жадқа деректерді жазады, ал екіншісі оларды оқиды.

Жады ортақ пайдаланатын 16 процессордың мультипроцессоры



Көп процессордағы барлық процессорлар бірдей мекенжай кеңістігін пайдаланатындықтан, операциялық жүйенің бір ғана көшірмесі жұмыс

істейді. Сәйкесінше, бір ғана жады бетінің картасы және бір процесс кестесі бар. Процесс блоктаған кезде оның процессоры өз күйін операциялық жүйе кестелерінде сақтайды, содан кейін басқа процесті

іске қосу үшін сол кестелерді іздейді. Дәл осы бір жүйеге негізделген ұйым

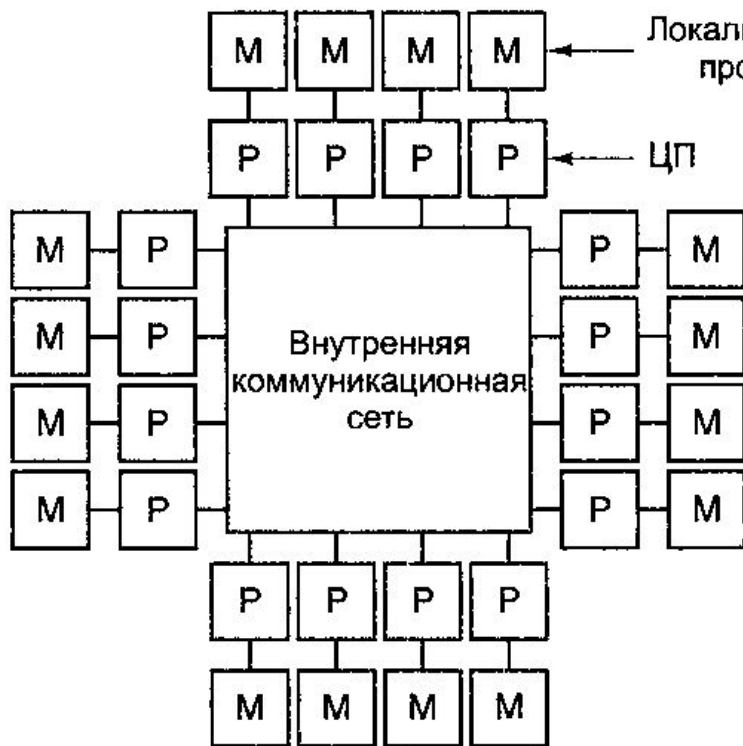
мультипроцессорды көп компьютерден ажыратады.

Көп процессордағы барлық процессорлар бірдей мекенжай кеңістігін пайдаланатындықтан, операциялық жүйенің бір ғана көшірмесі жұмыс істейді. Сәйкесінше, бір ғана жады бетінің картасы және бір процесс кестесі бар. Процесс блоктаған кезде оның процессоры өз күйін операциялық жүйе кестелерінде сақтайды, содан кейін басқа процесті іске қосу үшін сол кестелерді іздейді. Дәл осы бір жүйеге негізделген ұйым мультипроцессорды көп компьютерден ажыратады.

Мультикомпьютерлер

- * Параллельді архитектураның екінші нұсқасында әрбір процессордың тек осы процессорға ғана қолжетімді өз жады бар. Мұндай схема мультикомпьютерлік, немесе таратылған жады жүйесі деп аталады. Мультикомпьютер мен мультипроцессордың негізгі айырмашылығы - мультикомпьютердегі әрбір процессордың өзінің жергілікті жады бар, бұл процессор ЖҮКТЕП және САҚТАУ нұсқауларын орындау арқылы қол жеткізе алады, бірақ басқа ешбір процессор осы нұсқауларды пайдаланып бұл процессордың жергілікті жадына қол жеткізе алмайды. . Осылайша, мультипроцессорларда барлық процессорлар ортақ бір физикалық мекенжай кеңістігі болады, ал мультикомпьютерде әрбір процессор үшін жеке физикалық мекенжай кеңістігі болады.

- * Мультикомпьютердегі процессорлар ортақ жадқа қарапайым кіру арқылы бір-бірімен байланыса алмайтындықтан, процессорлар оларды байланыстыратын байланыс желісі арқылы хабарламалармен алмасады. Аппараттық құралда іске асырылатын ортақ жады болмаған жағдайда, белгілі бір бағдарламалық құрал құрылымы қабылданады. Көп компьютерде LOAD және STORE командаларымен ақпаратты оқуға және жазуға мүмкіндік беретін барлық процессорлар үшін бірыңғай виртуалды мекенжай кеңістігінің болуы мүмкін емес. Мысалы, жоғарғы сол жақ бұрыштағы процессор объектінің бір бөлігі келесі процессорға қатысты басқа бөлімге түсетінін анықтаса, ол ұшақтың құйрығының суретін алу үшін жай ғана жадтан ақпаратты оқуды жалғастыра алады.



а



б

Мультикомпьютерде жіберу және қабылдау примитивтері көбінесе процессорлар арасындағы байланыс үшін пайдаланылады.

Сондықтан

мультикомпьютерлік бағдарламалық қамтамасыз ету

көппроцессорлық

бағдарламалық қамтамасыз етуге қарағанда күрделі құрылымға

ие. Бұл

жағдайда негізгі мәселе деректерді дұрыс бөлу және оларды

орынды

орналастыру болып табылады. Бұл мультикомпьютер мен

мультипроцессордың тағы бір айырмашылығы, мұнда деректерді

орналастыру өнімділікке әсер етуі мүмкін болса да, мәселені

шешудің

дұрыстығына әсер етпейді.

Осылайша, біз дилеммаға тап болдық: мультипроцессорларды жобалау қиын, бірақ бағдарламалау оңай, ал мультикомпьютерді құру оңай, бірақ бағдарламалау қиын. Нәтижесінде гибриді жүйелерді

құру әрекеттері үнемі жүргізілуде. Бұл әрекеттер ортақ жадты әр опцияның артықшылықтары мен кемшіліктері бар әртүрлі тәсілдермен жүзеге асыруға болатынын түсінуге әкелді. Параллельді компьютер архитектурасы саласындағы барлық дерлік зерттеулер екі жүйенің де

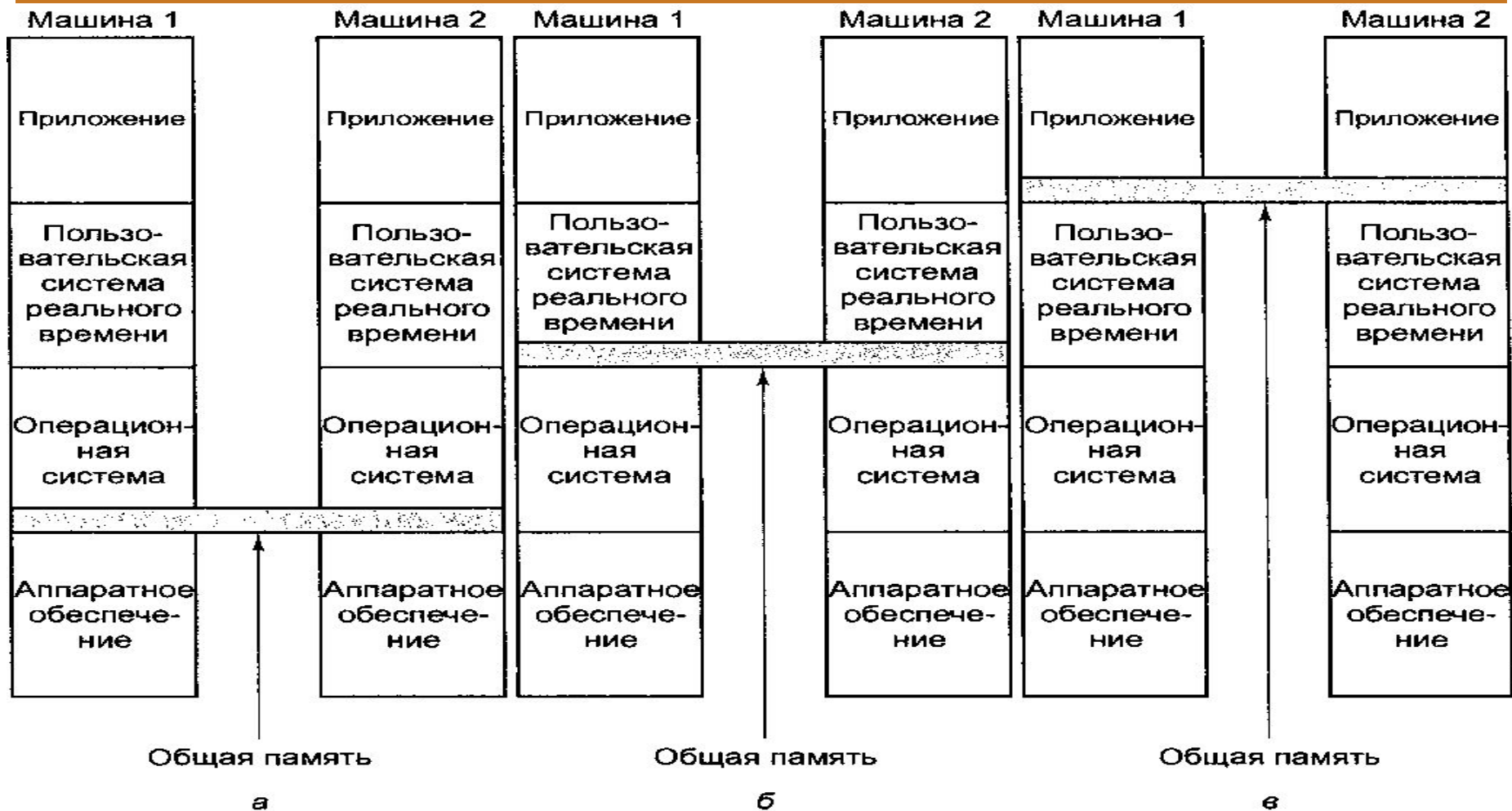
артықшылықтарын біріктіретін гибриді формаларды жасауға бағытталған.

Бұл жерде масштабтауға қол жеткізу маңызды, яғни процессорларды көбірек

қосу арқылы дұрыс жұмыс істейтін жапғастыратын жүйені дамыту

№1 Көзқарас

- * Бір тәсіл Қазіргі компьютерлік жүйелер монолитті емес, бірақ көп деңгейлі құрылымға ие екендігіне негізделген. Бұл суретте көрсетілгендей бірнеше деңгейдің кез-келгенінде жалпы жадты жүзеге асыруға мүмкіндік береді. 8.17. - Сур. 8.17, және біз "нақты" мультипроцессордағыдай аппараттық түрде енгізілген жалпы жадты көреміз. Бұл әзірлеуде бір кесте жиынтығы бар операциялық жүйенің бір көшірмесі, атап айтқанда жадты бөлу кестесі бар. Егер процесс көп жадты қажет етсе, ол операциялық жүйенің жұмысын тоқтатады, содан кейін ол кестеден бос бетті іздей бастайды және осы бетті қоңырау шалу процесінің мекен-жай кеңістігіне көрсетеді. Амалдық жүйеге келетін болсақ, бір жад бар және операциялық жүйе қай бетке жататынын бақылайды. Жалпы жадты аппараттық жолмен жүзеге асырудың көптеген жолдары бар.



№2 Көзқарас

- * бір виртуалды мекен-жай кеңістігін беру арқылы ортақ жадты модельдейтін көпкомпы және операциялық жүйені пайдаланыңыз. Бұл тәсілмен таратылған Жалпы жад алынады (таратылған ортақ жад, DSM), онда әр бет жад модульдерінің бірінде орналасқан, ал әр машинада өзінің виртуалды жады және жеке парақ кестелері бар [127]. Егер процессор LOAD немесе STORE пәрменін ол жоқ параққа сілтеме жасай отырып орындаса, жүйелік ерекшелік пайда болады

№3 Көзқарас

- * жалпы жадты нақты уақыттағы бағдарламалық жасақтама жүйесімен іске асырыңыз. Бұл тәсілде жалпы жадтың абстракциясы бағдарламалау тілін жасайды және бұл абстракцияны компилятор жүзеге асырады (яғни, жалпы жад моделі қолданылатын бағдарламалау тіліне байланысты болуы мүмкін). Мысалы, Linda моделі жалпы түйін кеңістігінің абстракциясына негізделген (өрістер жиынтығы бар деректер жазбалары). Кез-келген машинаның процестері ортақ кеңістікті алып тастай алады немесе оны ортақ кеңістікке жібере алады. Бұл кеңістікке кіру бағдарламалық жасақтамамен (нақты уақыт Linda жүйесі) толық басқарылатындықтан, арнайы аппараттық қолдау немесе арнайы операциялық жүйе қажет емес.



Соңы