10. Компютърни архитектури: Структура и йерархия на паметта. Сегментна и странична преадресация. Система за прекъсване – приоритети и обслужване.

Структура на основната памет. Йерархия – кеш, основна и виртуална памет. Сегметна и странична преадресация – селектор, дескриптор, таблици и регистри при сегметна преадресация; каталог на страниците, описател, стратегии на подмяна на страниците при странична преадресация. Система за прекъсване – видове прекъсвания, структура и обработка, конкурентност и приоритети, контролери на прекъсванията.

Структура на основната памет.

Памет в персоналния компютър се нарича всеки ресурс, който има свойството да съхранява информация във времето. Паметта представлява съвкупност от битове. Под един бит разбираме количество информация, за която отговаря един електронен елемент. Битът е минималната порция информация, която се съхранява, и има 2 стойности – 0 и 1. С развитието на компютрите се оказва, че най-удобната адресуема единица е байтът. Един байт е осем бита. Капацитетът на паметта се мери в байтове. Никоя памет концептуално не може да чете част от байт. Паметта е организирана в клетки, които са наредени последователно една след друга, т.е. като едномерен линеен масив от байтове, които се номерират с последователни номера, наречени адреси. Номерацията започва от 0 и стига до последния наличен физически адрес.

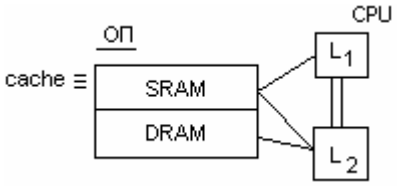
Паметта се изгражда като устройство, което може да извършва следните две операции:

- четене – подаване на адрес и извеждане съдържанието на съответния байт;
- писане – подаване на адрес и на байт, който се записва на този адрес.

Това е логическият модел на паметта. От архитектурна гледна точка паметта се реализира физически като тримерни матрици. Когато се подаде двоичен адрес, има устройство, което го дешифрира – адресът насочва устройството към онези елементи, които съдържат съответните битове. Времето за дешифрация и времето за прочитане са крайни времена. Под време за достъп до паметта разбираме времето от подаването на адреса до извличането на данните. Времето за достъп нараства с обема на паметта.

Паметта е организирана в йерархия от гледна точка на нейния обем и скорост на достъп до информация от нея. Тя се разбива на два големи класа: **основната памет** (RAM, Random Access Memory = SRAM + DRAM) и външната памет (която се организира върху магнитни ленти, магнитни барабани, магнитни дискове, CD, DVD и същевременен печат).

SRAM (Static RAM) памет, наречена още **Cache** памет – идеята е да се осъществява много бърз достъп до най-често използваните данни и команди, които да се съхраняват в кеш паметта, намираща се много близо до микропроцесора. Кешът изцяло се управлява от хардуера.



Кешът е на L1 и на L2 ниво. L1 нивото на кеша днес представлява неразделна част от централния процесор. Това не са регистрите на процесора. L2 на кеш паметта е извън ЦП и регистрите му, но достъпът между L1 и L2 е милиони пъти по-бърз, отколкото този между L2 и DRAM паметта. Тоест L1 е малка по обем памет, много бърза и много скъпа. L2 е по-

голяма по обем. Тя е по-бавна от L1 и сравнително евтина. Днес L2 паметта се прикрепя дори към управлението на самите устройства, като видеопамет, В/И система и т.н.

DRAM (Dynamic RAM) се реализира чрез една транзисторна схема, докато SRAM – с шест. DRAM с времето носи затихване на електрическия заряд и се налага през определен квант от време да се извиква т. нар. „опресняване на паметта". То е действие, което се предизвиква от кварцов итератор на квантови импулси и води до възобновяване на силата на електрическите сигнали. При SRAM такова опресняване не се налага.

Йерархия – кеш, основна и виртуална памет.

Тази схема показва **йерархията** на паметта. Колкото един вид памет е по-близо до върха, толкова тя е по-бърза, т.е. времето за достъп до нея е по-малко, но в същото време е и по-скъпа. Затова и, колкото е по-ниско в йерархията една памет, толкова е по-голяма по обем.

До регистрите има паралелен достъп от процесора. L1 кешът е разделен на две независими части – в едната се пазят само инструкции, в другата само данни. L2 кешът се намира в процесора и е част от общото адресно пространство, с което работят инструкциите. L3 кешът е извън процесора и е по-голям по обем от L2 кеша. Основната памет не съдържа всички адреси от общото адресно пространство, реално всички адреси могат да се поместят само върху диска.

Йерархията на паметта се базира на общото положение, че за малък период от време са нужни малко адреси. Нейната задача е да се реализира бърза памет чрез двете концепции за локалност:

- темпорална локалност – най-вероятно последно използваните данни ще бъдат използвани пак скоро, така всяко ниво помни последно използваните блокове от по-долното ниво;

- пространствена локалност – най-вероятно съседите в адресното пространство на последно използваните данни ще бъдат използвани скоро, така блоковете от по-ниските

нива в йерархията са по-големи, тъй като, освен че обхващат последно използваните блокове, те обхващат техните съседи.

В началото основната памет е много скъпа, докато магнитните дискове, макар по-бавни, са много по-евтини за запомняне на един бит. От друга страна програмите започват да нарастват и да се нуждаят от по-големи адресни пространства. Затова се въвежда принципът на **виртуалната памет** – работи се с виртуален адрес, който по размер е по-голям от физическия адрес на основната памет. Така виртуалното адресно пространство може да е много по-голямо по размер от реалното адресно пространство.

Странична преадресация – каталог на страниците, описател, стратегии на подмяна на страниците.

Основната концепция на виртуалната организация на паметта (ВОП) се състои в разграничаването на пространството на разработка на програма от пространството на адресите на реалната памет. Идеята е паметта за разработване на програма – виртуално адресно пространство (ВАП) – да е неограничена. От друга страна КС има една реална памет (различна за отделните модули) – физическо адресно пространство (ФАП). ВАП се разделя на фрагменти с еднакви размери, които наричаме **страници**. ФАП се разделя на фрагменти със същия размер, наричани блокове.

Апаратно или от операционната система се поддържа една таблица – **таблица на страниците**, която показва в даден момент от времето какво е изображението на виртуалната памет върху физическата, т.е. съдържа връзките между виртуалните и физическите адреси. В нея за всяка една от страниците на ВАП има по един ред, който съдържа информация за страницата, като например бит за наличност (ако е 1, то тази страница е разположена някъде в основната памет и това къде някъде се разполага в полето за номер на блок), както и dirty/modified bit (бит, който показва дали страницата е била променяна или не).

Виртуалният адрес се състои от номер на виртуалната страница и отместване в рамките на страницата. Процесът на преобразуване на виртуален адрес към реален физически адрес наричаме транслация. При транслацията номерът на виртуалната страница се преобразува в номер на физическа страница в реалната памет, отместването във физическата страница е същото като във виртуалната. Най-скорошно използваните записи от таблицата на страниците се кешират в ЦП, за да се оптимизира процеса по транслация.

Виртуалната памет има следните предимства:

- програмите могат да се изпълняват даже, когато не всичкият им програмен код или даннни са във физическата памет;
- защита – процесите са изцяло отделени един от друг;
- програмирането се опростява, тъй като всеки процес се развива в хомогенно виртуално адресно пространство с начален адрес 0.

Основен недостатък на виртуалната памет – какво става при ненамиране на физическата страница в паметта. Характерното за виртуалната памет е, че на диска се отделя файл (swap-file), в който се помества образ на пълното виртуално адресно пространство. В реалната памет се поместват само някои активни страници. При ненамиране липсващата страница трябва да се прехвърли от диска в реалната памет. Това прехвърляне е твърде

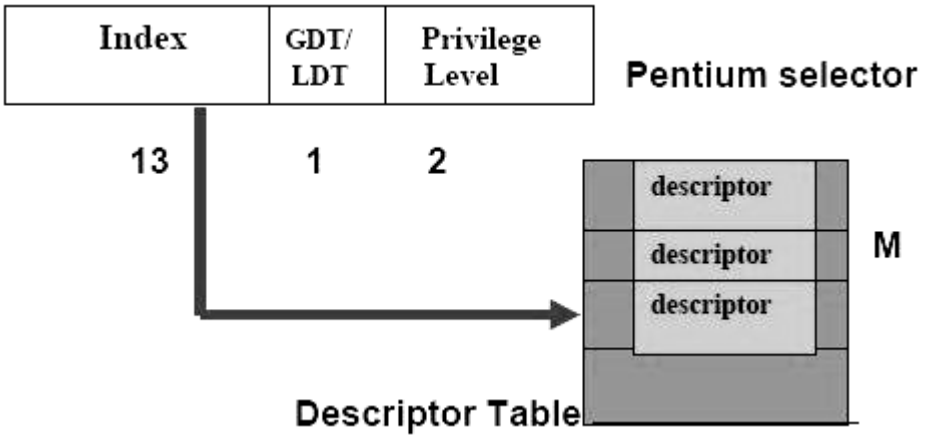
бавно и поради тази причина реалната памет винаги е write-back, т.е. никога от процесора не се записват данни в диска, така се поддържа връзка само между процесора и реалната памет.

Транслацията се извършва по следния начин: при заявка за достъп до дадена страница, първо се проверява дали тя се намира във физическата памет. Ако да – връща адреса на страницата. Ако не – зарежда търсената страница от диска и след това връща нейния адрес. При това, ако няма място в паметта за новата страница, някоя от наличните страници трябва да бъде подменена; също така, тъй като основната памет е write-back, ако подменяната страница е била променяна, тя трябва да се запише върху диска.

Сегметна преадресация – селектор, дескриптор, таблици и регистри.

Освен на страници, виртуалната памет може да бъде организирана и на **сегменти**, които за разлика от страниците са с променлива дължина и могат да се припокриват. Всеки сегмент представлява своето собствено адресно пространство. Един сегмент се състои от два компонента: базов адрес (адресът на някакво местоположение във физическата памет) и дължина (дължината на сегмента). Адресът на сегмента също се състои от два компонента: селектор на сегмента и отместване в сегмента. В машините PENTIUM при сегментацията адресът се получава от 16-битов сегментен селектор и 32-битово отместване. Всеки сегментен селектор съдържа индекс, показващ отместване в глобална или локална дескрипторна таблица (вж. по-долу за дескрипторните таблици), както и бит, показващ дали се използва ГДТ или ЛДТ.

За да бъде използван, един сегментен селектор трябва да бъде зареден в някой сегментен регистър (CS, DS, ES, SS).



Дескрипторните таблици съдържат сегментни дескриптори, като всеки сегментен дескриптор описва един сегмент чрез адреса на началото на сегмента, размера на сегмента и различни битове за състояние. Глобалната дескрипторна таблица (ГДТ) е единствена за системата и се използва най-вече за дескриптори на системни сегменти. Локалната дескрипторна таблица (ЛДТ) не е единствена, има по една за всеки процес и се използва най-вече за дескриптори на приложни сегменти. В даден момент може да се използва точно една локална дескрипторна таблица. ГДТ, освен сегментни дескриптори, може да съдържа и други неща, например дескриптори на ЛДТ. По идея една ЛДТ трябва да съдържа дескриптори на сегменти, които се отнасят до една конкретна програма, докато ГДТ съдържа глобални дескриптори.

При сегментацията, получаването на адреса по зададени сегментен селектор и отместване става по следния начин:

- първо се определя в ГДТ или ЛДТ да се търси сегментния дескриптор (в зависимост от бита GDT/LDT в сегментния селектор);
- след като е определен сегментният дескриптор, се взима адресът на началото на сегмента и към него се прибавя отместването.

Ако отместването е по-голямо от дължината на сегмента, системата сигнализира за грешка, т.е. за опит за нарушаване на сегментната защита.

Полученият адрес след сегментацията се нарича линеен адрес и той е част от линейното адресно пространство.

Възможни са два случая:

- линейното адресно пространство директно се изобразява върху физическото;
- линейното адресно пространство е виртуално – извършва се странична преадресация.

В случай на странична преадресация, линейният адрес се изпраща към блок, който го преобразува към физически адрес.

Система за прекъсване – видове прекъсвания, структура и обработка, конкурентност и приоритети, контролери на прекъсванията.

Прекъсването е процес, при които процесорът прекратява нормалното изпълнение на дадена програма, съхранява необходимата информация в стека и преминава в някакъв предварително избран адрес на паметта. След обработката на извиканата процедура, управлението се връща в изходната точка и продължава изпълнението на първоначалната програма. **Системата за прекъсване** цели във всеки момент да се даде възможност за регистрация на случващо се събитие. Механизмът на прекъсванията е ефективен начин за обмяна на информация с бавните ВИ устройства и за сигнализиция за особени състояния в работата на централния процесор. Чрез нея се избягва необходимостта от периодични проверки на флагове за дадени събития. На всяко прекъсване се съпоставя точно определен номер. За всеки вид прекъсване има специална програма, която се изпълнява при възникването на този вид прекъсване. В основната памет има специална фиксирана област, наречена таблица на векторите на прекъсванията. Всеки вектор на прекъсване съдържа указател към подпрограма за обработка на прекъсването и състояние. Тези указатели са разположени на фиксирано място в ОП (адреси от 0 до 1023 – по 4 байта за всеки от 256-те указателя).

Когато възникне прекъсване, ЦП прекъсва изпълнението на текущата програма и съдържанието на програмния брояч PC и на регистъра на състоянието MSW се запазват в стек, за да може да има вложени прекъсвания. Новото състояние на процесора при влизане в прекъсване се взима от съответния вектор на прекъсване, който се намира по номера на прекъсването. Всяка програма, която обслужва прекъсване, завършва с инструкция за връщане от прекъсване, която от върха на стека прочита предишното състояние на PC и MSW и ги зарежда в процесора и по този начин изпълнението се връща към прекъснатата програма.

Съществуват различни **видове прекъсвания**:

- по машинна грешка – грешка в апаратурата, те са най-високо приоритетни;
- входно-изходни прекъсвания – те се активират в резултат на изпълнение на входно-изходна операция;
- външни прекъсвания – свързани са с някакъв външен елемент (например бутон RESET) и са аналогични на входно-изходните прекъсвания;
- програмни прекъсвания – те са синхронни с програмата, която се изпълнява и подтискат изпълнението на някаква инструкция, например деление на 0;
- програмно-активирани (SVC) прекъсвания – при тях текущата програма издава специална команда за провеждане на прекъсване.

Друго разделяне на прекъсванията е на маскируеми и немаскируеми. Маскируемите прекъсвания могат да се игнорират, т.е. да не се обработват, докато немаскируемите прекъсвания задължително се обработват. Така немаскируемите прекъсвания винаги имат приоритет пред маскируемите.

За да се отчете важността на събитията, които може да настъпят, се въвежда **приоритет** на прекъсванията. По време на изпълнението си програмата за обработка на едно прекъсване може да бъде прекъсната само от прекъсване с по-висок приоритет. Обикновено прекъсванията с по-малки номера имат по-висок приоритет.

Ако едновременно са възникнали няколко прекъсвания, те се обработват в следния ред:

- прекъсвания от инструкцията INT и при особени случаи;
- прекъсване при стъпков режим (debugging mode);
- немаскируемите прекъсвания;
- маскируемите прекъсвания.

Управлението на прекъсванията се извършва от специална логическо устройство, което се нарича **контролер на прекъсванията**. Контролерът на прекъсванията получава заявки за прекъсвания от различните хардуерни устройства посредством линиите за заявки за прекъсвания IRQ (Interrupt Request – заявка за машинно прекъсване). Машинното прекъсване е нужно, за да се избегнат хаотичните заявки към процесора на различни външни устройства, нуждаещи се от управление или обмяна на данни, при които се губи много процесорно време. IRQ позволява бърза реакция при устройства, които се нуждаят от бърза обработка, за да не губят информация. Естествено IRQ има и недостатъци, например дадена програма може да забрани машинните прекъсвания за определено време.

По-елементарно обяснено ще изглежда така: контролерът на прекъсванията приема заявките за прекъсване на външните устройства, определя приоритетите и след това изпраща заявка за прекъсване към процесора, т.е. контролерът „решава" кое устроиство, кога и колко време ще комуникира с процесора, като ги редува. Например: видео картата е „хванала" и работи с IRQ11, това означава, че тя ще обменя информация по линия 11, след нея е примерно модемът, които ще обменя по IRQ12, после звуковата карта и т.н. Част от IRQ са резервирани за стандартни устройства на дънната платка (например IRQ0 - System timer; IRQ1 - Keyboard и т.н.), а друга част са на разположение на различни външни устройства, включени на шината (видео карта, звукова карта, вътрешен модем и др.).