6. Структура и йерархия на паметта. Сегментна и странична преадресация. Система за прекъсване – приоритети и обслужване.

Анотация: Йерархия на паметта – кеш-памет, оперативна памет и виртуална памет.

Сегментна преадресация
- – сегментен селектор;
- – сегментен дескриптор;
- – сегментни таблици и регистри.

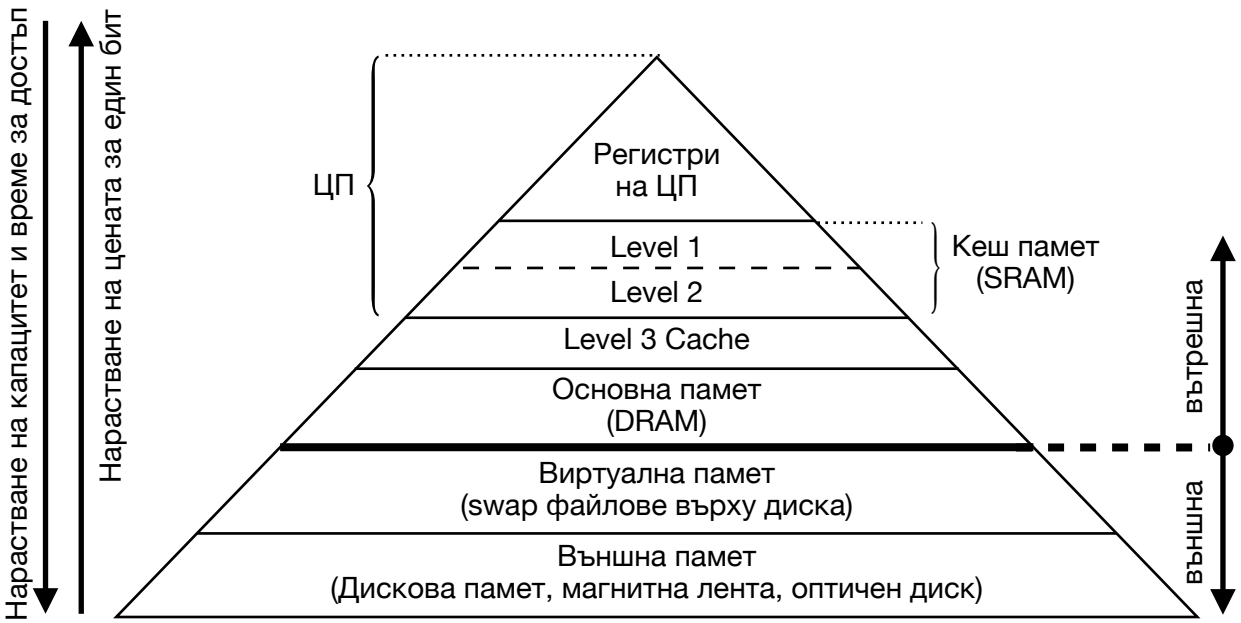
Странична преадресация
- – каталог на страниците;
- – описател на страница;
- – стратегии на подмяна на страниците.

Прекъсвания
- – структура и обработка;
- – типове прекъсвания;
- – конкурентност и приоритети;
- – обслужване на прекъсванията;
- – контролери на прекъсванията.

Паметта е мястото в един персонален компютър, където се съхраняват данни, които се използват от програмите. Тя представлява съвкупност от битове. Един бит представлява отговор на логически въпрос с два възможни отговора (0 или 1) и е най-малката мерна единица за измерване на количество данни (информация). С развитието на компютрите се оказва, че най-удобната адресируема единица е байтът. Един байт се състои от 8 бита. Капацитетът на паметта се измерва именно в байтове.

Йерархия на паметта

Паметта може да се раздели на няколко йерархии, като всяко ниво има различна скорост за достъп, цена за бит и големина на капацитета. Процесора може да използва всяко едно от тях според своите нужди. Йерархията на паметта може да се представи чрез фигурата по-долу.



фиг. 1. Диаграма на йерархия на паметта

Тази йерархия на паметта е разделена на два основни раздела:
- **Външна памет (Вторична памет)** – състои се от магнитния диск (или така наречения твърд диск), оптичен диск и магнитна лента. Това са видове памет, които са достъпни от процесора чрез някакъв входно-изходен модул;
- **Вътрешна памет (Първична памет)** – състои се от основната памет, кеш памет и регистри. Тази памет може директно да бъде достъпвана от страна на процесора.

Характеристиките, които се променят в зависимост от това на кое ниво се намира дадената памет са:

- **Капацитет** – мярка за това колко най-много байтове може да съхранява дадения вид памет. Капацитетът нараства с нарастването на нивата;
- **Време за достъп** – времето, за което една операция за четене или писане ще достъпи парче памет от дадено ниво;
- **Цена за един бит** – колкото по-високо е нивото на паметта, в което се намира дадения бит, толкова по-висока е цената му.

Кеш памет
Кеш паметта е специална високоскоростна памет. Тя се използва за забързването на операциите по четене от паметта от страна на ЦП. Именно заради това кеш паметта е по-скъпа отколкото например дисковата памет (и заради това капацитетът ѝ е по-малък), но за сметка на това кеш паметта е в пъти по-бърза за достъп от дисковата памет. Кеш паметта е по-бърза, защото запазва най-често или най-скоро използваните данни и инструкции. По този начин следващия път, когато същите данни или инструкции се поискат от ЦП, те няма да бъдат извлечени отново от оперативната или дисковата памет, а директно от кеш паметта, която се намира най-близо до процесора.

Кеш паметта от своя страна се дели на 3 нива: Level 1, Level 2 и Level 3 (L1, L2 и L3).
L1 кешът е най-бързият за достъп от всички нива, но за сметка на своята бързина, той е и най-малкият, като варира между 8 и 64 килобайта (1 килобайт = 1024 байта). Това ниво кеш се дели на две части – кеш с инструкции и кеш с данни. Кешът с инструкции съхранява инструкции за операцията, която процесора изпълнява в момента. Кешът с данни от друга страна съхранява данните, върху които работи самата операция.

L2 и L3 кешовете са по-бавни, но пък за сметка на това те разполагат с много повече капацитет, като Level 3 може да достигне до мегабайти памет (1 мегабайт = 1024 килобайта). От тук следва, че колкото по-големи са тези два кеша, толкова концептуално по-бързо може да работи един компютър.

Всяко ядро на даден процесор си има собствени L1 и L2 кеш, като L3 е споделен между всички ядра.

Cache hit/miss
Когато процесорът иска да прочете дадени данни от паметта, той първо проверява в кеш паметта. Ако той намери нужните му данни там, то тогава настъпва cache hit събитие, което обозначава, че операцията по извличане от кеша е била извършена успешно. Това събитие се отчита, за да бъде използвано после в изчисляването на съотношението на попаденията и всички заявки към кеша (hit ratio). Ако обаче, процесора не намери данните в кеша, то тогава се регистрира събитие cache miss (пропуск). Когато това настъпи, кеша се обновява, като иска от оперативната памет стойностите на търсените данни и след като се обнови процесора извлича данните директно от кеша. Съотношението на попаденията (hit ratio) се изчислява със следната формула:

$$\text{Hit ratio} = \frac{\text{hits count}}{\text{hits count} + \text{miss count}}$$

Тя ни дава оценка за това колко добре се представя кеш паметта. Според нейната стойност могат да се правят различни видове оптимизации върху кеша, целящи да повишат това съотношение.

Оперативна памет
Оперативната памет съдържа инструкциите за централния процесор и данните, върху които тези инструкции работят. Функцията на оперативната памет в днешно време се изпълнява от RAM паметта, като не е изключено и кеш паметта да е част от оперативната памет.

RAM паметта е съкращение от Random Access Memory (памет с произволен достъп). Когато се извършват операции по писане или четене във външната памет, операциите ще имат променливо време за изпълнение, тъй като зависят от самото местоположение на данните. Произволността в името на RAM паметта идва от това, че всяка операция по писане или четене ще има едно и също време независимо от местоположението на данните. Това е и причината RAM паметта да е по-бърза за достъп, отколкото външната памет. RAM паметта се счита за променлива, тъй като тя губи своето състояние, когато системата загуби ток.

RAM паметта се дели на два типа: SRAM (Static RAM) и DRAM (Dynamic RAM). Статичната RAM памет се използва изцяло за кеш памет, която обяснихме по-горе. Динамичната RAM памет се използва като основна памет. Други разлики между двата типа RAM памети са например:

- SRAM използва цели шест транзистора, за да съхранява данните, докато DRAM използва една транзисторна схема;
- Структурата на SRAM е значително по-сложна от тази на DRAM;
- Тъй като SRAM е кеш памет, то тя е по-скъпа и по-бърза от DRAM.

Виртуална памет
Виртуалната памет е вид съхранение на данни, при който външната памет може да бъде адресирана така сякаш е част от основната памет. Когато една компютърна система достигне до състояние, в което RAM паметта ѝ е била изчерпана, тя създава виртуална памет в част от външната си памет и ѝ прехвърля част от RAM паметта, за да я освободи. Виртуалната памет е временна и тя предимно се използва, когато RAM паметта трябва да бъде изчистена, за да може да изглежда така сякаш компютърната система да може да използва повече оперативна памет, отколкото това физически е възможно. Адресирането към части от данните върху виртуалната памет се осъществява чрез виртуални адреси. За да може тези данни реално да се извличат (тъй като виртуалните адреси не сочат към правилното физическо място в паметта) се осъществява транслация на виртуален адрес към машинен (физически). Тъй като самата виртуална памет се намира върху външната памет, то извличането на данните от нея е скъпа операция откъм време.

Странична преадресация
Основната концепция при виртуалната памет е разграничаването на пространството на разработка на програма от пространството на адресите на реалната програма. Идеята е паметта за разработване на програма (или виртуално адресно пространство – ВАП) да е неограничена. Но от друга страна, всеки компютър има свое собствено физическо адресно пространство (ФАП). ВАП се разделя на фрагменти с еднакви размери, които се наричат страници, а ФАП също бива разделен на такива фрагменти, които наричаме блокове.

Таблица на страниците
Операционната система поддържа таблица, която наричаме таблица на страниците или каталог. Този каталог ни показва в даден момент от времето какво е изображението на виртуалната памет върху физическата, тоест 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 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 същата страница. При транслация на виртуалния адрес към машинен (физически) номерът на виртуалната страница се преобразува в номер на физически блок в реалната памет, а отместването не се променя. Най-скорошно използваните записи от каталога се кешират, за да се оптимизира процеса по транслация.

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. Ако подменената страница е била променяна, то тя трябва да се запише върху диска.

Сегментна преадресация
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, като той се състои от два компонента: базов адрес (адрес към физическата памет) и дължина (дължина на сегмента). Адресът на сегмента също се състои от два компонента: селектор на сегмента и отместване в сегмента. Всеки сегмент съдържа индекс, показващ отместване в глобалната или локалната дескрипторна таблица, както и бит, показващ дали се използва глобална или локална дескрипторна таблица.

За да бъде използван, един сегментен селектор трябва да бъде зареден в някой сегментен регистър.

Дескрипторните таблици съдържат сегментни дескриптори, като всеки един описва един сегмент чрез адреса на началото на сегмента, размера му и различните битове за състояние. Глобаната 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 ЛДТ. Освен сегментни дескриптори, ГДТ може да съдържа и други компоненти като например дескриптори на ЛДТ. ГДТ съдържа глобални дескриптори, докато ЛДТ съдържа дескриптори на сегменти, които се отнасят до една конкретна програма.

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

- Първо се определя дали да се търси сегментния дескриптор в ГДТ или ЛДТ.
- След като е определен сегментния 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.

Прекъсвания
Прекъсването е процес, при който 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 входно-изходни устройства и за сигнализация за особени състояния в работата на ЦП. Чрез тази сигнлизация 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 оперативната памет.

Когато възникне прекъсване, ЦП прекъсва изпълнението на текущата програма и съдържанието на програмния брояч на регистъра на състоянието се запазва 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 програмния брояч и регистъра на състоянието и ги зарежда в процесора и по този начин ипълнението 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 и са аналогични на входно-изходни прекъсвания;
- Програмни прекъсвания – те са синхронни с програмата, която се изпълнява и подтискат изпълнението на някаква инструкция, например деление на 0;
- Програмно-активирани (SVC – Supervisor Call) прекъсвания – при тях текущата програма издава специална команда за провеждане на прекъсване.

Маскируеми и немаскируеми прекъсвания
Друго разделяне на прекъсванията е на маскируеми и немаскируеми. Маскируемите прекъсвания могат да се игнорират, тоест 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 (debuggin mode)
- Немаскируемите прекъсвания;
- Маскируемите прекъсвания.

Управлението на прекъсванията се извършва от специално 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 което се губи много процесорно време. IRQ позволява бърза реакция при усройства, които се нуждаят от бърза обработка, за да не губят информация. Естествено IRQ има и недостатъци, напримр дадена програма може да забрани машинните прекъсвания за определено време. Казано по-просто, контролерът на прекъсванията приема заявките за прекъсване на външните устройства, определя приоритетите и след това изпраща заявка за прекъсване към процесора, тоест контролерът „решава“ кое устройство, кога и колко време ще комуникира с процесора, като ги редува.