

Виртуална памет

I. Въведение

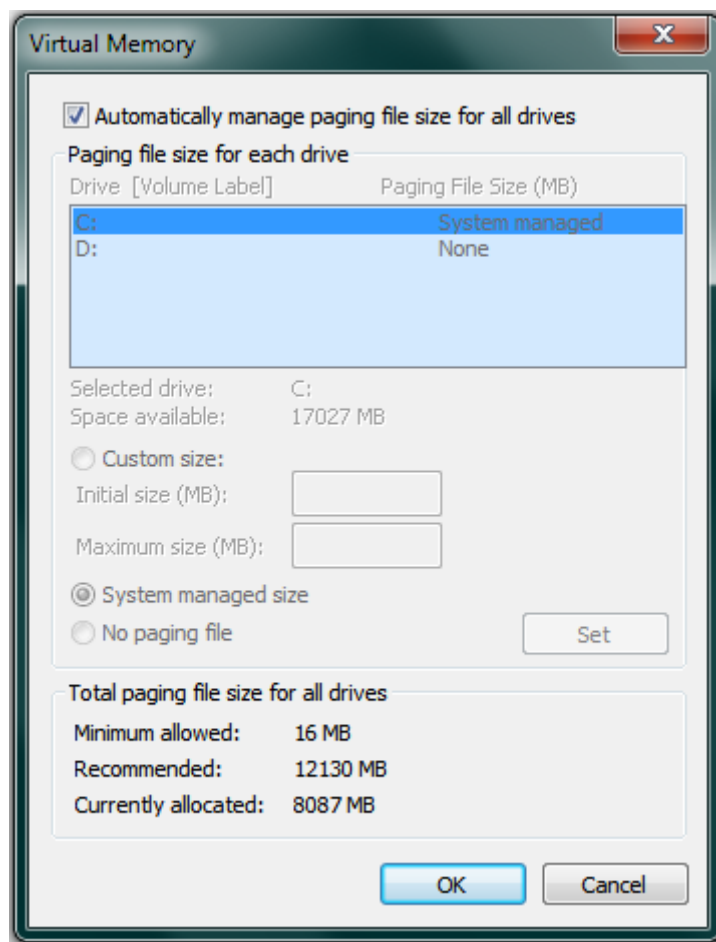
Виртуалната памет е техника за управление на паметта, която осигурява "идеализирана абстракция на ресурсите за съхранение, които действително са достъпни на даден компютър", което създава илюзията на потребителите за много голяма (безкрайна) физическа памет. Идеята е проста – при недостиг на физическа оперативна памет да се използва по-евтината дискова памет. Тази идея е предложена при разработването на компютър с кодово име Atlas в Университета на град Манчестър в края на 50-те години на миналия век.

Необходимостта от виртуална памет е в пряко следствие на използването на много-задачни и / или многопотребителски операционни системи. При тях вероятността за недостиг на физическа оперативна памет в даден момент от време е голяма. Използването на виртуална памет позволява на тези операционни системи да се стартират множество процеси, като техния максимален брой зависи не от количеството на инсталираната оперативна памет, а от типа на операционната система и заделената от нея дискова памет, която ще се използва като виртуална. Не е достатъчно операционната система да може да работи с виртуална памет – необходимо е и микропроцесора да поддържа това.

Основният недостатък на използването на виртуална памет е забавеното изпълнение на чакащите за обработка процеси. Причината за това са бавните дискови операции (четене и запис) с цел достъпване на виртуалната памет. Следователно, добро решение е виртуалната памет да е разположена в дял на SSD диск. Виртуалната памет заема фиксирано място на някой от дяловете на диска, по подразбиране главния дял. Тази част от диска не е достъпна за несистемен достъп, следователно не подлежи на четене, запис, преместване и изтриване от приложни програми. Най-често това е системен файл, който има специфична организация, специфична за всяка операционна система.

Какво точно се записва във виртуалната памет? При недостиг на физическа оперативна памет операционната система трябва да изтрие информацията за даден процес, за да може да се зареди данни и програмен код, който се налага да се стартира (рестартира). В този случай във виртуалната памет се налага да се буферира само контекста на процеса. Програмният код не е необходимо да се буферира, тъй като операционната система знае кой е изпълнимия файл, който го съдържа. Контекстът на един процес се формира от данните на процеса и съдържанието на всички програмно достъпни регистри на микропроцесора в момента когато процесът трябва да бъде прекъснат с цел преместване във виртуалната памет. Получаването на контекста, буферирането му във виртуалната памет, както и възстановяването му при рестартиране на процеса се реализира с участието на микропроцесора. Целта е намаляване на времето за превключване на контекста. За да е възможно това е необходимо микропроцесорът да има необходимите регистри, за да адресира специфични структури, използвани за превключване на контекста. Тези регистри и структури са различни при различните микропроцесори.

На Фиг. 1 е показано разпределението и възможностите за конфигуриране на виртуалната памет при операционна система Windows 7.

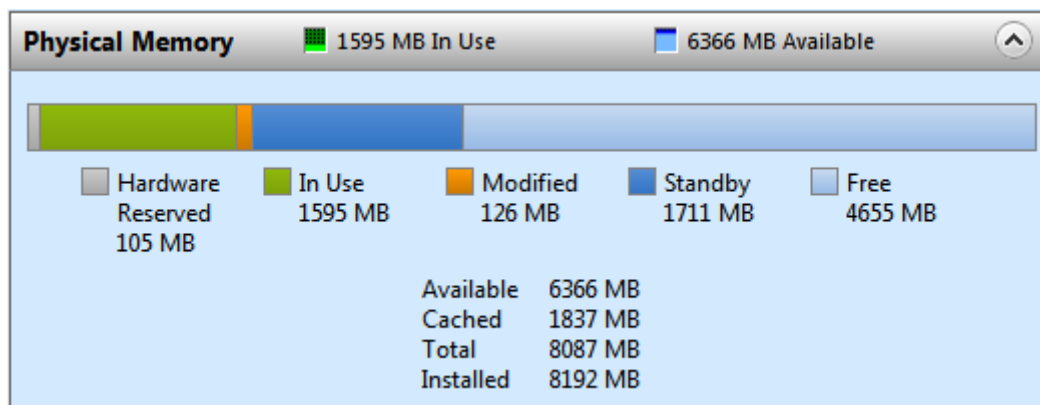


Фиг. 1. Конфигуриране на виртуалната памет при ОС Windows 7

При конкретния пример е зададено автоматично конфигуриране на виртуалната памет. В този случай файлът, който изпълнява функцията на виртуална памет (paging file), се разполага в главния дял (диск C:). Размерът на този файл се задава като процент от инсталираната физическа оперативна памет. В този случай инсталираната физическата памет 8 GiB от които разпределени са $N=8087$ MiB (виж Фиг. 1). Препоръчителния размер на paging файла (при конкретната операционна система) е $K_1 \times N$, където $K_1=1.5$ (12130 MiB), а максималния размер е $K_2 \times N$, където $K_2=3$. При Windows 10 препоръчителният размерът на този файл зависи от достъпната физическа памет, но коефициентът K_1 не е константа, например: $K_1 = 1.25$ при $N = 8$ GiB; $K_1 = 2.5$ при $N = 16$ GiB; $K_1 = 5$ при $N = 32$ GiB.

От Фиг. 1 се вижда, че можете да зададете друго разпределение за виртуалната памет, различно от това по подразбиране. За целта изключете автоматичното разпределение и изберете дял в който да се разположи paging файла. В този случай може да зададете собствен размер на този файл. Съществува опция за забраняване на използване на paging файл (No paging file). Трябва да се има предвид, че повечето операционни системи използват виртуалната памет, дори това да е забранено и дори да имате достатъчно количество инсталирана физическа памет, например над 64 GiB.

На Фиг. 2 е показано моментното разпределение на физическата оперативна памет. Инсталираната памет е 8 GiB (8192 MiB). От тях са достъпни 8087 MiB – 105 MiB са резервирани за системни нужди. От тях са кеширани 1837 MiB, а 6366 MiB са достъпни за използване. От тези 6366 MiB са свободни 4655 MiB, а се използват 1595 MiB.



Фиг. 2. Моментно разпределение на физическата оперативна памет

II. Управление на оперативната и виртуалната памет

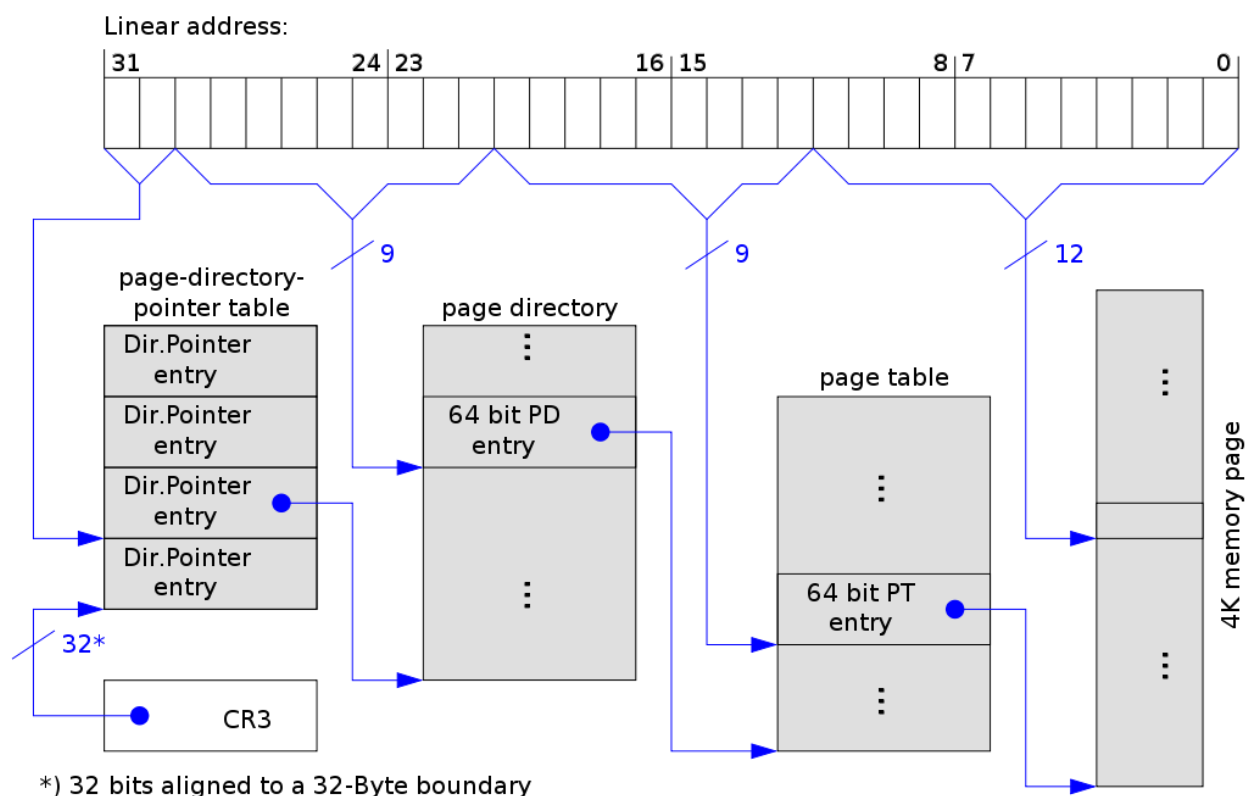
Всяка операционна система има собствена стратегия за управление на оперативната памет. Тази стратегия се изпълнява от модула за управление на паметта (memory management). Най-често паметта се разделя на блокове (страници) всеки от които има собствен манипулатор (идентификатор). Всеки блок има статус – свободен или зает. Само операционната система (или процес с root права) може да чете и записва в блоковете от паметта. Най-малката единица информация, която може да бъде преместена към / от виртуалната памет е една страница с данни. С течение на времето остават множество непоследователни свободни блокове в паметта, които не се използват от нито един процес. Управлението на блоковете, както и преместването им с цел обединяване на малките свободни блокове с цел получаване на по-голям свободен блок е възможно само от ядрото на операционната система. Този процес се нарича фрагментиране на оперативната памет. Всяка операционна система има собствена стратегия чрез която намалява фрагментацията. Тази стратегия не може да се активира много често, тъй като по време на дефрагментиране се блокира изпълнението на процесите.

По аналогичен начин се получава и фрагментиране на блоковете с които се работи във виртуалната памет. Операционните системи рядко се ангажират с дефрагментиране на paging файловете, тъй като това е много бавна операция. По-често се заделя по-голям размер за paging файла, за да се не се налага да се дефрагментира съдържанието му.

III. Адресиране на виртуалната памет

Като всяка памет, така и виртуалната памет трябва да може да се адресира. Съвременните процесори не адресират директно оперативната памет, тъй като нямат регистри с необходимия за това размер. Те използват техники за индиректно адресиране на оперативната памет. Най-често използваните подобни техники са сегментиране и страницирание. Повечето от операционните системи за персонални компютри управляват паметта чрез страницирание. В този случай паметта се разделя на страници с размер 4KB или 2MB. За всяка страница могат да бъдат задавани отделни права за достъп. Всяка

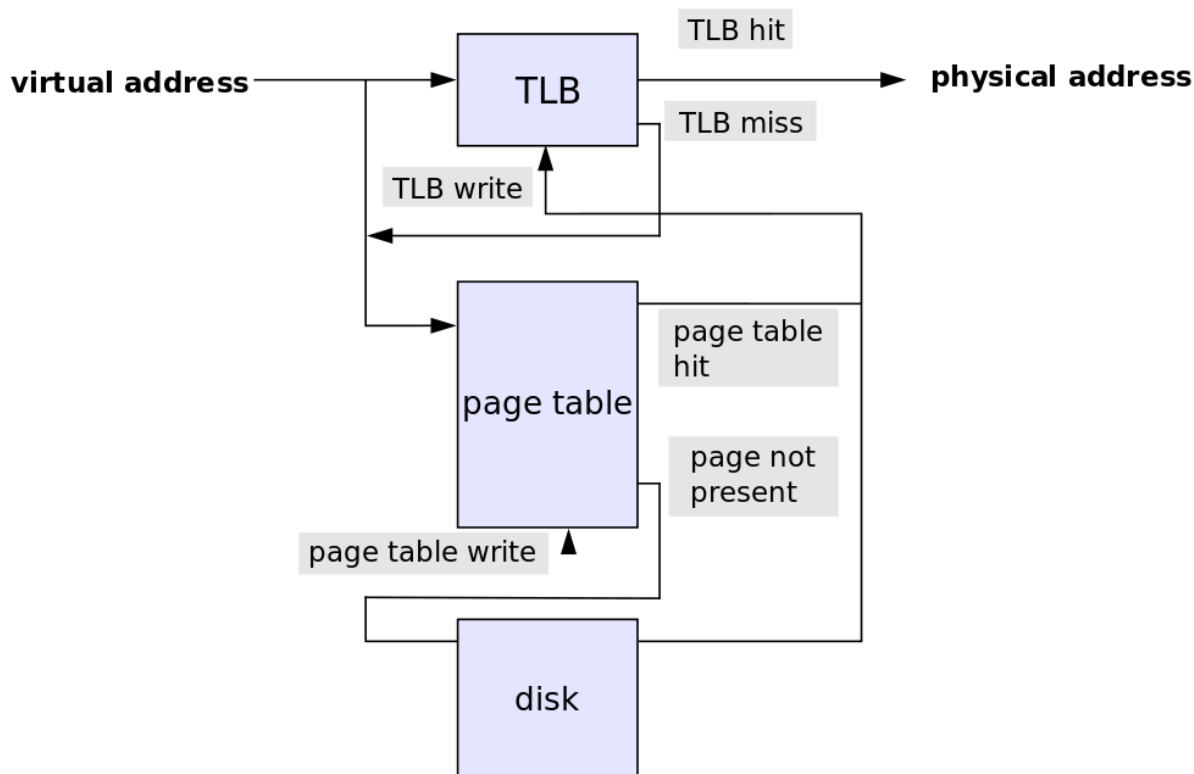
страница може да се намира във физическата памет или във виртуалната памет. На фиг. 3 е показан начина на адресиране на 4KB страница при процесор на Intel.



Фиг. 3. Адресиране на физическата оперативна памет при 4 KiB страници

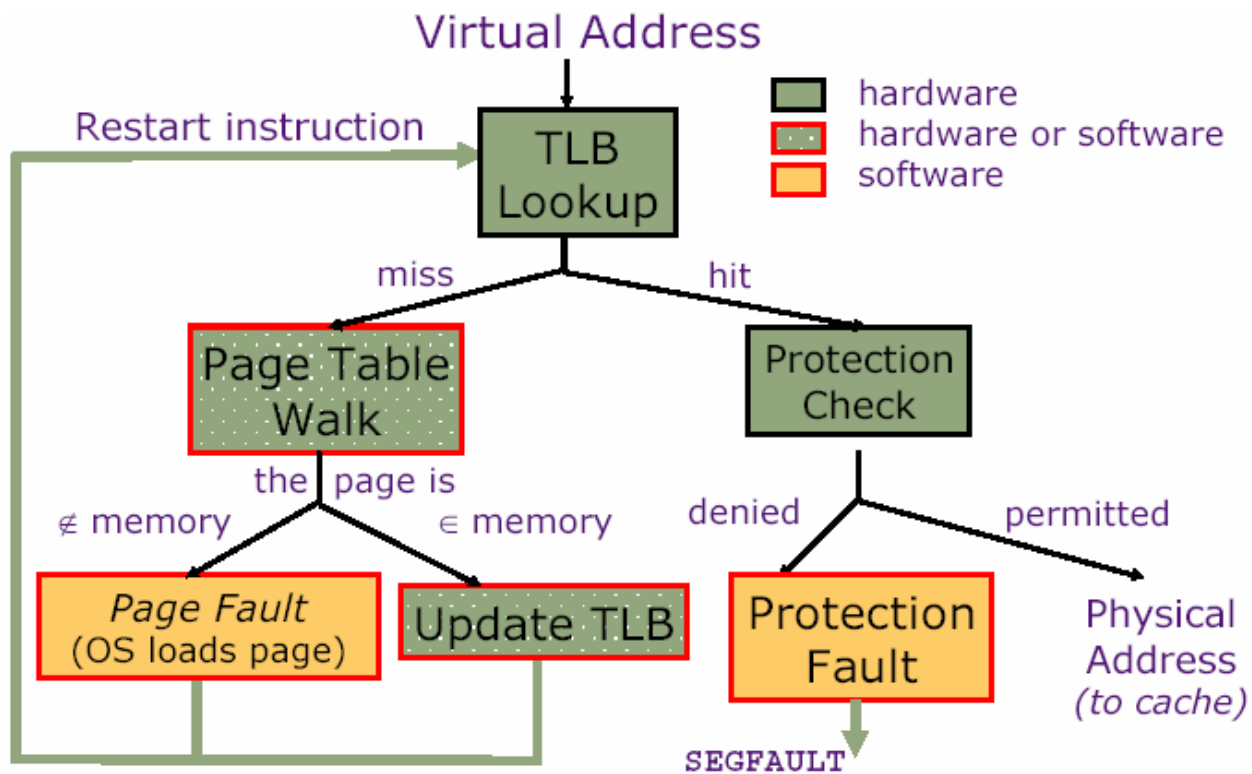
Линейният (виртуален) адрес, който микропроцесорът генерира, се разделя на 4 части. Старшите 2 бита са отместване в структура наречена Page Directory Pointer Table (PDPT). Началото на тази структура се сочи от текущото съдържание на Управляващ Регистър 3 (CR3). Всеки елемент от PDPT е указател към началото на друга структура наречена Page Directory (PD). Всеки елемент от PD е указател към структура с име PageTable (PT). Всеки елемент от PT (PTE) е дескриптор на една страница в паметта. Отместването в страницата се задава чрез младшите 12 адреса на линейния адрес.

След като се адресира страница от паметта трябва да се провери дали тя се намира във физическата оперативна памет или във виртуалната памет. За целта, в дескриптора на всяка страница има бит с име Present (P). Ако този бит е 1, то страницата е във физическа памет. За да се ускори конвертирането на линейните до физически адреси се използва асоциативна кеш памет в която се буферират последно конвертираните адреси. Тази памет се нарича Translation Lookaside Buffer (TLB). На Фиг. 4 е показано как се използва TLB при процесорите на Intel. Когато виртуален (линеен) адрес трябва да бъде преобразуван до физически адрес, първо се проверява дали в TLB има запис за този линеен адрес. Ако се намери съвпадение (TLB hit), от TLB се извлича физическия адрес и достъпът до паметта може да продължи с кеширане на паметта. Ако обаче няма съвпадение (TLB miss), се активира търсене на физическия адрес в таблицата на страниците (Page Table) при 4KB страници или в PD при 2MB страници.



Фиг. 4. Преобразуване на виртуален до физически адрес при работа с виртуална памет

Ако дадена страница се адресира, но не е налице (page not present), ще възникне събитие от тип Page Fault и операционната система трябва да го обработи. В този случай се налага страницата да бъде намерена във виртуалната памет и заредена обратно във физическата памет (виж Фиг. 5).



Фиг. 5. Събития, генерирани при работа с виртуалната памет

Ако физическият адрес вече е буфериран в TLB се реализира проверка дали текущият процес има права за достъп до паметта (Protection Check), която той сочи. При липса на права се генерира събитие Protection Fault, което операционната система трябва да обработи. Следователно, операционната система се ангажира само с обработката на грешки при достъп до паметта, а всичко останало се поема от микропроцесора.

V. Въпроси и задачи за самостоятелна работа

1. Кога се налага използването на виртуална памет? Има ли операционни системи при които тази памет не се използва?
2. Защо по подразбиране paging файла е разположен в главния дял на диска?
3. Обосновете поради каква причина бихте преместили paging файла от главния дял в друг дял на диска.
4. Използвайте специализирана литература за да получите съдържанието на дескрипторите на страниците при процесори на Intel и AMD. Какви битове, освен бит Present има в тези структури и за какво се използват те?
5. Как може да получите размера на TLB?
6. Като използвате специализирана литература проверете как операционна система Windows 10 управлява оперативната памет.