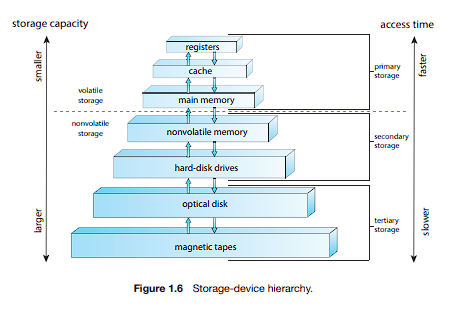
**سیستم عامل جلسه ششم**

**سلسه مراتب حافظه**

بین سه ویژگی کلیدی حافظه یعنی ، هزینه ، ظرفیت، زمان دسترسی باید سبک و سنگین کرد. برای این کار نمی توان بر یک حافظه یا فن آوری خاصی تکیه کرد و باید از سلسله مراتب حافظه استفاده کرد

****

**سلسله مراتب متدوال حافظه:**

1. ثبات ها (Register)
2. حافظه نهان (cache)
3. حافظه اصلی (Main Memory)
4. حافظه پنهان دیسک
5. دیسک مغناطیسی (Hard Disk - HDD)
6. رسانه جا به جا پذیر (USB – DVD - CD)

با حرکت به سطوح پایین تر این سلسله مراتب شرایط زیر رخ می دهد:

1 – کاهش هزینه در هر بیت 2- افزایش ظرفیت 3- افزایش زمان دسترسی

4- کاهش تعداد دفعات دسترسی پردازنده به حافظه

**مدیریت منابع هر یک از حافظه های زیر در مقابل آن نوشته شده است:**

ثبات ها: کامپایلر

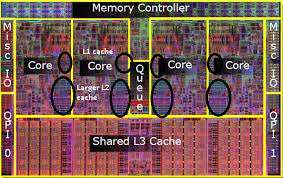
حافظه پنهان: خودکار (توسط خود پردازنده)

حافظه اصلی و دیسک : سیستم عامل

**حافظه نهان (cache)**

A CPU cache is a hardware cache used by the central processing unit (CPU) of a computer to reduce the average cost (time or energy) to access data from the main memory. A cache is a smaller, faster memory, located closer to a processor core, which stores copies of the data from frequently used main memory locations.

یک حافظهی کوچک و سریع بین پردازنده و حافظه اصلی (Main Memory) است. حافظه پنهان حاوی بخشی از حافظه اصلی است.



وقتی پردازنده می خواهد کلمه ای از حافظه را بخواند وجود آن را در حافظه ی پنهان بررسی می کند. اگر وجود داشته باشد به پردازنده تحویل داده می شود در غیر این صورت یک بلوک از حافظه اصلی شامل تعداد ثابتی از خانه های حافظه به حافظه پنهان منتقل می شود و سپس کلمه مورد نظر به پردازنده تحویل داده می شود. هنگامی که یک بلوک از داده ها به حافظه پنهان آورده می شود تا یک مراجعه به حافظه انجام شود ، به دلیل پدیده ی محلی بودن مراجعات ، احتمالا به زودی به دیگر کلمات آن بلوک نیز مراجعه خواهد شد.

در یک سیستم کامپیوتری، برای افزایش کارایی از حافظه های چند سطحی استفاده می کنند به گونه ای که سطوح نزدیک تر به پردازنده دارای ظرفیت کمتر اما در عوض سرعت بیشتری هستند. یکی از این موارد استفاده از حافظه پنهان است بدین صورت که هرگاه دستوراعمل یا داده ای در حافظه اصلی مورد استفاده قرار گیرد یک کپی از آن در حافظه پنهان ایجاد می شود . دلیل آن این است که بر اساس اصل محلی گرایی گفته می شود هر گاه داده هایی مورد استفاده قرار گیرند به زودی در آینده نیز لازم خواهد بود. قرار دادن این داده ها و دستوراعمل ها در حافظه پنهان موجل افزایش سرعت دستیابی می شود بنابراین زمانی که داده یا دستوراعملی نیاز باشد ابتدا به حافظه پنهان مراجعه می شود و در صورت وجود آن، استفاده می شود.

در غیر این صورت به حافظه اصلی مراجعه شده و داده ها و دستوراعمل های مورد نظر، مورد استفاده قرار می گیرند.

از آن جایی که ظرفیت حافظه پنهان نسبت به حافظه اصلی بسیار کمتر است بنابراین نخواهیم توانست همه داده هایی را که در حافظه اصلی هستند را به حافظه پنهان ببریم پس ممکن است گاهی به داده هایی نیاز داشته باشیم که در حافظه پنهان نیستند. اگر چنانچه حافظه پنهان پر شده باشد و نیاز به خالی کردن بخشی از آن و جایگزینی آن با داده مورد نظر داشته باشیم از الگوریتم های جایگزینی حافظه استفاده می شود که دقیقا همان الگوریتم های جایگزینی صفحه در حافظه اصلی هستند

یادداشت:

صفحه (Page) در حافظه اصلی (RAM) به عنوان یک بخشی از فضای آدرس‌پذیر قرار دارد و به نوعی یک بلاک داده است که شامل اطلاعاتی مانند برنامه‌ها، داده‌ها و سایر اطلاعاتی است که توسط سیستم عامل به حافظه اصلی بارگذاری می‌شود. هر صفحه، دارای یک سایز خاص است و به طور معمول، اندازه یک صفحه در حافظه اصلی، 4 کیلوبایت است.

هنگامی که یک برنامه را اجرا می‌کنید، سیستم عامل صفحات مربوط به برنامه را در حافظه اصلی بارگذاری می‌کند. در فرآیند اجرای برنامه، اگر برنامه به داده‌های جدیدی نیاز داشت، صفحات جدید به صورت پویا در حافظه اصلی ساخته می‌شوند و اگر برنامه به صفحات قبلی دسترسی نداشت، صفحات قبلی از حافظه اصلی حذف می‌شوند.

استفاده بهینه و مناسب از صفحات در حافظه اصلی، باعث افزایش سرعت و کارایی سیستم شما می‌شود. بنابراین، طراحی و پیاده‌سازی الگوریتم‌هایی برای مدیریت صفحات در حافظه اصلی، یکی از اصلی ترین وظایف سیستم عامل است

**سخت افزایه پایه (Base Hardware)**

حافظه اصلی و ثبات های ساخته شده در خود پردازنده ، تنها فضای ذخیره سازی همه منظوره ای هستند که پردازنده مستقیما می تواند به آن ها دسترسی داشته باشد

هر دستور العمل در حال اجرا و داده ای که توسط آ« استفاده می شود، بای در یکی از این دو دستگاه ذخیره سازی با دستیابی مستقیم (حافظه ی اصلی و ثبات ها) واقع باشد.

**انقیاد آدرس (Address Binding)**

معمولا برنامه بر روی دیسک به صورت یک فایل اجرایی دودویی ذخیره می شود. برنامه باید به حافظه بار شود و در داخل فرایندی قرار گیرد تا اجرا شود. بر حسب این که چه مدیریت حافظه ای مورد استفاده قرار می گیرد. این فرایند ممکن است در حین اجرا بین دیسک و حافظه انتقال یابد. فرایند های موجود در دیسک که منتظرند وارد حافظه و اجرا شوند، صف ورودی را تشکیل می دهند.

**فضای آدرس منطقی و فیزیکی (physical and logical address)**

**یادداشت:**

**A logical address is the virtual address that is generated by the CPU. A user can view the logical address of a computer program. On the other hand, a physical address is one that represents a location in the computer memory. A user cannot view the physical address of a program.**

آدرسی که توسط پردازنده تولید می شود، آدرس منطقی (Logical Address) نام دارد، درر حالی که آدرسی که توسط واحد حافظه مشاهده می شود (آدرسی که به ثبات آدرس حافظه بار می شود)، آدرس فیزیکی (Physical Address) نام دارد

مجموعه ای از تمام آدرس های منطقی که توسط برنامه ای تولید می شود، فضای آدرس منطقی (Logical Address Space) نام دارد. مجموعه ای از تمام آدرس های فیزیکی متناظر با این آدرس های منطقی، فضای آدرس فیزیکی نام دارد.

نگاشت زمان اجرا از ادرس های مجازی به فیزیکی، توسط واحد مدیریت حافظه (MMU) Memory Management Unitانجام می شود که یک دستگاه سخت افزاری است.

برنامه ی کاربر هیچ گاه آدرس های فیزیکی واقعی را نمی بیند.

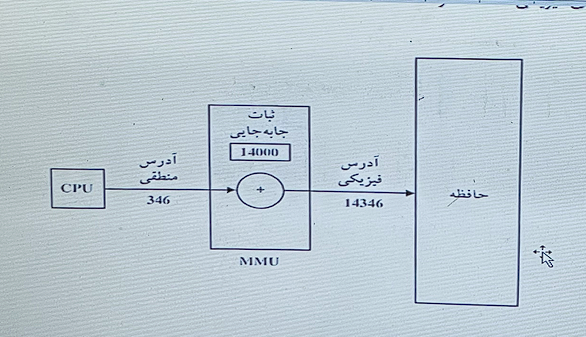
سخت افزار نگاشت حافظه ، آدرس های منطقی را به آدرس های فیزیکی تبدیل می کند

دو نو آدرس وجود دارد

آدرس های منطقی (از صفر تا max)

آدرس های فیزیکی (از 0 + R تا max + R با مقدار پایه ی R )

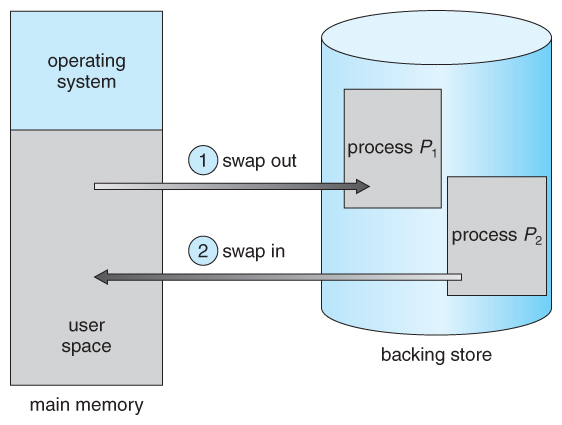
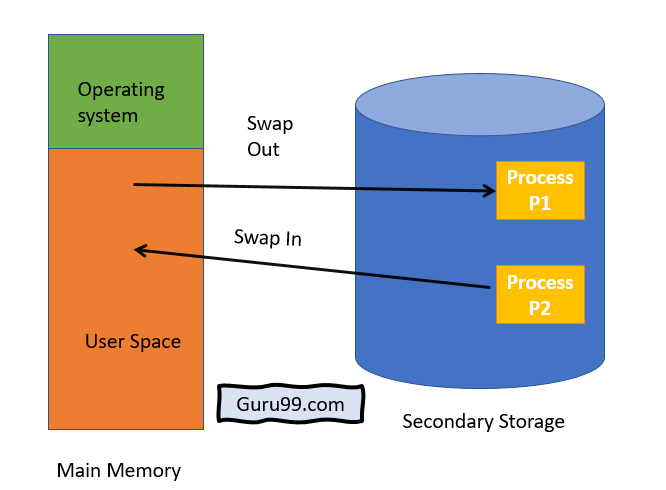
کاربر فقط آدرس های منطقی را تولید می نماید و فکر می کند که فرایند در محل های صفر تا max اجرا می شود. برنامه کاربر ، آدرس های منطقی را تولید می کند و این آدرس ها قبل از بکارگیری باید به آدرس های فیزیکی نگاشت شوند.



**مبادله (Swapping)**

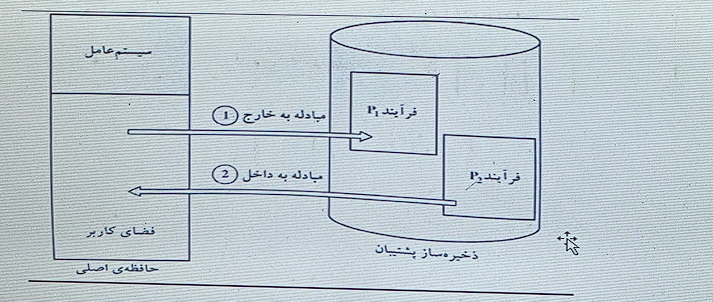
فرایند باید در حافظه باشد تا اجرا شود. اما، فرایند می تواند موقتا از حافظه اصلی به ذخیره ساز پشتیبان برود و بعد به حافظه برگردد و به اجرایش ادامه دهد. این عمل را مبادله می گویند

Swapping is a technique used in operating systems to temporarily move pages or entire processes from the main memory (RAM) to secondary storage devices, such as hard disks, to free up space in the main memory. When the system needs to access the data that has been swapped out, it swaps the less frequently used data back into the main memory and swaps out other data to secondary storage again. This process is called swapping because the system is essentially swapping data between the main memory and secondary storage. Swapping enables a system to run larger applications or more applications concurrently than it could if it relied solely on the available physical memory.

**مبادله استاندارد**

مبادله ی استاندارد شامل انتقال فرایند بین حافظه ی اصلی و ذخیره ساز پشتیبان است. ذخیره ساز پشتیبان معمولا یک دیسک سریع است و باید به اندازه کافی بزرگ باشد تا بتواند تمام تصاویر و حافظه را برای تمام کاربران ذخیره کند و دستیابی مستقیم به این ذخیره ساز را فراهم آورد.

****

**تخصیص حافظه ی همجوار (Contiguous Memory allocation)**

حافظه ی اصلی باید سیستم عامل و فرایند های کاربران را جا دهد. بنابراین لازم است بخش های مختلفی از حافظه اصلی به روش کارآمدی تخصیص یابند.

حافظه معمولا به دو قسمت تقسیم می شود:

یک قسمت برای سیستم عامل مقیم

قسمت دیگر برای فرایند های کاربر

یادداشت:

تخصیص حافظه‌ی همجوار یک روش اختصاص حافظه است که در آن، برای یک فرآیند یا برنامه، یک بلوک پیوسته و پی در پی از حافظه به طول مشخصی اختصاص داده می‌شود. در این روش، مسئولیت تخصیص و اداره حافظه به عهده‌ی سیستم عامل است.

استفاده از حافظه‌ی همجوار در برنامه‌ها باعث بهبود کارایی و سرعت اجرای برنامه می‌شود، زیرا با این روش، زمان لازم برای جستجوی فضاهای خالی در حافظه کاهش می‌یابد و برنامه به راحتی می‌تواند به سمت قسمت‌های پیوسته از حافظه حرکت کند.

اما با این حال، تغییر در اندازه یا نوع حافظه‌ای که به یک برنامه اختصاص داده شده است، ممکن است سبب شکستگی برنامه شود و مشکلاتی برای سیستم عامل و پردازنده‌ها ایجاد کند. بنابراین، استفاده از تخصیص حافظه‌ی همجوار در صورتی مناسب است که برنامه به طور دقیق نیازهای خود را مشخص کرده باشد و نیاز به تخصیص و انتقال پویا در حین اجرای برنامه نداشته باشد.

سیستم عامل را می توان در آدرس بالای حافظه یا در آدرس پایین حافظه قرار داد. عامل مهمی که در این تصمیم گیری نقش دارد، محل بردار وقفه (Interrupt vector-table) است. هر سطح وقفه یک محل رزرو شده در حافظه دار که بردار وقفه نامیده می شود. چون معمولا بردار وقفه در آدرس پایین حافظه قرار دارد ، برنامه نویسان معمولا سیستم عامل را نیز در آدرس پایین حافظه قرار می دهند.

**تخصیص حافظه (Memory Allocation)**

یکی از ساده ترین روش ها برای تخصیص حافظه این است که حافظه را به چندین بخش (پارتیشن) با اندازه ی ثابت تقسیم شود. در هر بخش ممکن است دقیقا یک فرایند قرار گیرد. لذا درجه ی چند برنامه ای توسط تعداد بخش ها محدود می شود.اگر در این روش چند بخشی، یک بخش از حافظه خالی باشد، فرایندی که از صف ورودی انتخاب می شود و در آن بخش خالی قرار میگیرد.وقتی فرایندی خاتمه می یابد، بخشی از حافظه که در اختیار آن است ازاد می شود و فرایند دیگری می تواند درآن قسمت قرار گیرد

در طرح بخش هایی با اندازه های متفاوت، سیستم عامل ، جدولی را تشکل می دهد که مشخص می کند چه بخش هایی از حافظه ، آزاد و چه بخش هایی اشغال هستند. در آغاز کل حافظه برای فرایند های کاربر مهیا است و به عنوان یک بلوک حافظه ی بزرگ به نام حفره (Hole) در نظر گرفته می شود. سرانجام حافظه شامل مجموعه ای از حفره ها با اندازه های مختلف خواهد بود

وقتی فرایند ها وارد سیستم می شوند، در یک صف ورودی قرار می گیرند. سیستم عامل نیازمندی های حافظه ی مربوط به هر فرایند و فضای موجود را در نظر می گیرد و مشخص می کند به کدام فرایند ها ، حافظه تخصیص دهد. وقتی به فرایندی حافظه تخصیص می یابد. به حافظه بار می شود و می تواند برای دریافت چرخه های پردازنده رقابت کند . وقتی فرایندی خاتمه می یابد. حافظه اش را آزاد می کند و سیستم عامل می تواند آن را به فرایند دیگر موجود در صف ورودی اختصاص دهد.

IMG-3635