

جهت خرید فایل word به سایت [www.kandooon.com](http://www.kandooon.com) مراجعه کنید  
یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۵۱۱ تماس حاصل نمایید



دانشگاه آزاد اسلامی

حافظه مجازی

استاد: .....

نام دانشجو: .....

## حافظه مجازی

در این تمرین شما با صورتها و شکل‌های مختلفی از مکانیزم حافظه مجازی در ویندوز NT آشنایی پیدا خواهید کرد علی‌رغم اغلب دیگر سیستم‌های عامل، ویندوز NT یک API ساده و روشنی را برای اداره کردن بعضی شکلها و صورتهای حافظه مجازی تهیه می‌کند(معمولاً حافظه مجازی بطور کامل توسط برنامه نویس کاربردی‌اش روشن و واضح می‌گردد) در این تمرین شما در موارد زیر اطلاعاتی یاد خواهید گرفت:

- ❖ سازماندهی سیستم حافظه مجازی ویندوز NT
- ❖ چگونه فضای حافظه مجازی خود را کنترل کنید؟
- ❖ چگونه یک وسیله آگاه کننده و گزارش دهنده بنویسید؟
- ❖ جزئیات - VirtualQuery - GetSystemInfo , GlobalMemory Status - VirtualUnlock - VirtualLock - VirtueaFree - VirtualAlloc

## معرفی

حافظه مجازی صفحه بندی یا Paging Virtual Memory در بسیاری از سیستمهای عامل امروزی بکار گرفته می شود. در یک سیستم صفحه بندی شده، هر فرآیندی یک فضای آدرس دهی مجازی خاص خود دارد که برای ارجاع دیگر اشیاء بکار گرفته می شود که معمولاً محتوای یک محل یا موقعیتی از حافظه است بخشی از فضای آدرس دهی مجازی توسط ویراستار خطی ( Link editor ) تعریف می شوند وقتی که آن یک تصویر قابل اجرایی بوجود می آورد که در واقع فایل اجرایی یا EXE است. تعداد باقیمانده از فضای آدرس دهی می تواند بطور پویا در زمان اجرا توسط روشهایی که در این تمرین توضیح داده خواهد شد تعریف شود. بعد از اینکه قسمت پایدار و ثابت از فضای آدرس دهی مجازی ایجاد شد در حافظه ثانویه ذخیره خواهد شد ( معمولاً در بخش یا Partition از وسیله ذخیره سازی که Paging disk نام دارد ). به منظور عملی تر شدن شما می توانید Paging disk را مشابه فایل در نظر بگیرید.

در یک کامپیوتر معمول و مرسوم پردازنده تنها می تواند دستورات را واکنشی کند یا داده هایی را که در حافظه اولیه یا قابل اجرا ( که معمولاً RAM خوانده می شوند. ) واقع

شده‌اند را بارگذاری کند. حافظه اولیه در مقایسه با حافظه ثانویه کوچکتر و سریعتر است. حافظه اولیه خیلی گرانتر از حافظه ثانویه است بنابراین بطور معمول در زمره اجزای با ارزش‌تر از نظر حجم اما کوچکتر از حافظه‌های ثانویه قرار دارند.

اغلب کامپیوترها حافظه اولیه کافی حتی برای ذخیره فضای آدرس‌دهی مجازی کامل یک فرآیند را هم ندارند بنابراین در یک زمان تعداد زیادی فضا روی حافظه ثانویه برای ذخیره فضای آدرس‌دهی مجازی تعداد زیادی فرآیند وجود خواهد داشت. حافظه اولیه همچنین خیلی سریعتر از حافظه ثانویه می‌باشد. پردازنده می‌تواند یک بایت را در ۲ سیکل پردازنده در حافظه اولیه بخواند یا بنویسد. اما همین عمل هزاران سیکل از پروسسور را برای نوشتن یا خواندن اطلاعات در حافظه ثانویه نیازمند است.

برای نگهداری فضای حافظه اولیه یک سیستم حافظه مجازی صفحه‌بندی شده در هر زمان داده شده تنها بخشی از فضای آدرس‌دهی مجازی تعداد مختلفی از فرآیندها را بارگذاری یا (Load) می‌کند. همانطور که Threadها در فضای آدرس‌دهی فرآیندهایشان اجرا می‌شوند بخشی از فضای آدرس‌دهی مجازی که در حال حاضر در حال استفاده است در حافظه اولیه بارگذاری می‌شود و در همان حال دیگر بخشهای فضای آدرس‌دهی در حافظه ثانویه قرار گرفته‌اند. زمانی که فرآیندی به بخشی از فضای آدرس‌دهی مجازی دیگر نیاز

ندارد ( حداقل برای مدتی ) از آن بخش در حافظه ثانویه کپی گرفته می شود. این به موقعیتی از حافظه اولیه که مورد استفاده برای ذخیره بخشی از فضای آدرس دهی مجازی قرار گرفته بود این اجازه را می دهد که برای ذخیره بخش دیگری از فضای آدرس دهی مجازی در زمان دیگری بکار گرفته شود.

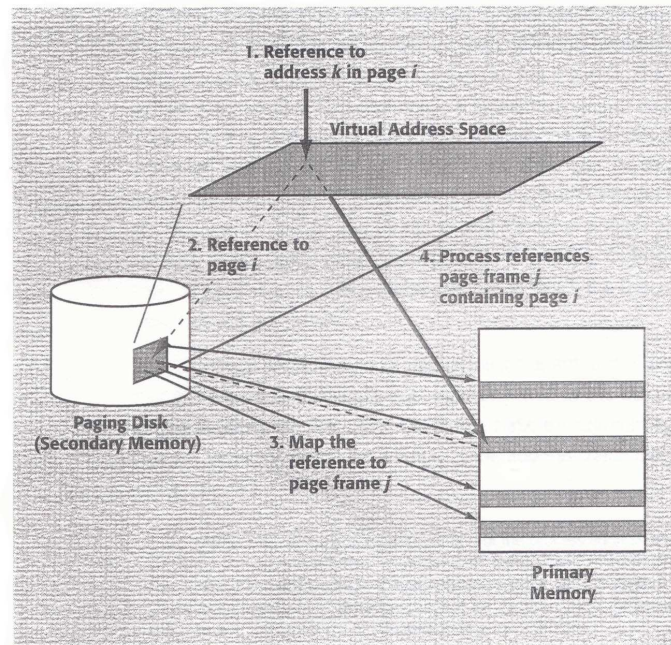
در یک سیستم حافظه مجازی سنجش در کارایی بوسیله کپی کردن یک بلوک از حافظه در حافظه اولیه یا بازگرداندن به حافظه ثانویه در زمانی که یک جابه جایی بین دو سطح از سلسله مرتبه حافظ مورد نیاز و ضروری باشد بدست می آید کارایی از این واقعیت بدست می آید که عملیات ورودی و خروجی حافظه ثانویه وابسته به بلوکها می باشد. این یعنی اگر تنها یک کلمه ( Word ) اطلاعات از حافظه ثانویه مورد نیاز باشد همه بلوک باید خوانده شود تا به آن کلمه برسیم. همچنین نسبت به مکانی که همه آن بلوک در حافظه اولیه قرار داد هم احساس است تا زمانی که باید خوانده شود تا به لغت گم شده برسیم.

فوائد دیگری در بارگذاری همه بلوکها نسبت به فقط یک کلمه وجود دارد. هنگامی که یک نخ یا Thread به مکانی چون I رجوع می کند احتمال زیادی وجود دارد که بخواند به مکان I+1 در آینده نزدیک رجوع کند به این مفهوم محلّیت یا Locality گفته می شود. یک حافظه مجازی صفحه بندی شده بلوکهای با اندازه ثابت را بارگذاری می کند و یا



برمی دارد که به آنها صفحه یا Page گفته می شود که در زمانی که داده ها و اطلاعات را بین حافظه اولیه و ثانویه در جهت رفت و برگشت حرکت می دهد انجام می گیرد. حد و مرز صفحه ها یا Page ها کاملاً توسط برنامه نویس مشخص و واضح می شود. شکل ۱ عملیات یک سیستم حافظه مجازی صفحه بندی شده بطور عام را خلاصه می کند.

زمانی که یک نخ یا Thread به آدرس مجازی K رجوع می کند (مرحله ۱ در شکل) حافظه مجازی ابتدا تعداد صفحات یا Page هایی که آدرس مجازی K را در بردارد تعیین می کند (مرحله ۲ در شکل) اگر صفحه در حال حاضر در حافظه اولیه بارگذاری شده بود و موجود بود (مرحله ۳ در شکل) سیستم حافظه مجازی آدرس مجازی را به آدرس فیزیکی متناظر با آن موقعیت در حافظه اولیه که همان Page Frame است تبدیل می کند (جایی که صفحه هدف در آن واقع شده است). اگر صفحه در همان زمان که به آن رجوع شده بارگذاری نشده بود اجرای Thread عادی دچار وقفه می شود تا زمانی که مدیریت حافظه مقصد را در Page Frame بارگذاری کند به محضی که آن صفحه بارگذاری شد اجرا ادامه پیدا خواهد کرد در مرحله ۴ رجوع به آدرس مجازی K دوباره با آدرس فیزیکی در حافظه اولیه تعیین می گردد (مکانی که موقعیت مجازی K در حال حاضر آنجا بارگذاری شده است).



شکل شماره ۱- سیستم صفحه بندی کلی

جنبه های مثبت یک سیستم حافظه مجازی این است که بسیاری از فرآیندها از حافظه اولیه در یک زمان استفاده می کنند حتی زمانی که جمع فضای آدرس دهی مجازی آنها خیلی بزرگتر از مقدار حافظه اولیه در کامپیوتر است. ارزش پشتیبانی از حافظه مجازی زمانی است که یک نخ یا thread در حال اجرا به حالتی برمی خورد که باید منتظر شود تا بخشی از فضای آدرس دهی مجازی اش در حافظه اولیه بارگذاری شود. حافظه مجازی در جایی که تعداد دفعات زمانی که thread یا نخ باید در انتظار بماند مکرر است (که بصورت کلاسیک به آن time

versus space trade off می گویند) چیز خوبی است. ویندوز NT سیستم حافظه مجازی صفحه بندی شده را مورد استفاده قرار می دهد. گونه های زیاد منحصر بفردی از حافظه مجازی ویندوز NT وجود دارد بنابراین در باقیمانده بحث به طراحی ویندوز NT متمرکز می شویم.

### مدل حافظه مجازی ویندوز NT

به هر فرآیند در ویندوز NT یک فضای آدرس دهی مجازی با اندازه ثابت ۴ گیگا بایت داده می شود که البته خیلی بزرگتر از مقدار حافظه اولیه (RAM) در کامپیوترهای شخصی (PC) معاصر آنها است. فرآیند نیاز ندارد که همه فضای آدرس دهی مجازی را استفاده کند تنها مقداری که احتیاج دارد را استفاده می کند. بطور عادی فایل های EXE برای یک برنامه خیلی کوچکتر از فضای آدرس دهی هستند. همانطور که در شکل ۲ توضیح داده شده است بخشی از فضای آدرس دهی مجازی که معمولاً ۲ گیگا بایت است بکار برده می شود تا به نخ یا thread اجازه دهد که به object های فضای حافظه کاربر رجوع کند و مقدار باقیمانده برای



رجوع آدرسهای که توسط سیستم عامل بکار برده می شود (Supervisor space) مورد استفاده قرار می گیرد.

اگر چه قسمت فضای ناظر یا همان Supervisor space از فضای آدرس دهی، در فضای آدرس دهی مجازی فرآیند وجود دارد اما تنها می توان بوسیله یک thread که در حال اجرا در مد ناظر است به آن رجوع کرد در وضعیت عادی اولین ۶۴ کیلو بیت (آدرس 0\*00000000 تا

0\*0000FFFF) از فضای آدرس دهی مجازی استفاده نمی شود. علت آن این است که

زمانی که برنامه نویس کدی با اشاره گری اشتباه می نویسد اشاره گر بد معمولاً صفر یا عددی

کوچک می باشد. در سیستم حافظه مجازی ویندوز NT اگر یک برنامه تلاش کند تا به هر

فضایی در ۶۴k اول حافظه رجوع کند سیستم حافظه مجازی فرض می کند که ارجاع یک

اشتباه برنامه ای بوده و اشتباه حافظه اتفاق افتاده است. اولین بخش قابل استفاده از فضای

آدرس دهی از 0\*00010000 شروع می شود و به 0\*7FFEFFFF ختم می شود. ۱۲۸k

کمتر از ۲GB بنابراین اولین مکانی که برنامه بکار می برد معمولاً 0\*00010000 است.

موقعیتهای 0\*7FFF0000 تا 0\*7FFFFFFF بعنوان اولین ۶۴k برای برنامه های فضای

kernel ذخیره و رزرو شده اند. علاوه بر اینکه بعنوان یک حفاظتی برای بدست آوردن

ارجاعات به آدرسهای کوچک است. از 0\*80000000 تا 0\*FFFFFFFF یک بخش 2 GB

کامل است که در آن راه اندازهای قطعات و هسته یا kernel سیستم و برنامه های قابل اجرا واقع شده اند سیستم عامل به چندین وسیله برای تعیین میزان فضای آدرسی که فرآیند واقعاً بکار می برد نیاز دارد همانطور که در بحث سیستم صفحه بندی کلی اشاره شد پیوند دهنده متن یا همان (link editor) تصویر اجرایی ثابتی در یک فایل EXE می سازد که معمولاً بخشی از فضای آدرس دهی که برای کدهای کامپایل شده مورد استفاده قرار میگیرد را تعریف می کند. DLL و دیگر قسمت های تخصیص پویایی فضای آدرس دهی می توانند به فضای آدرس دهی مجازی در زمان اجرا اضافه شوند.

۲ مرحله برای اضافه کردن آدرس بطور پویا به فضای آدرس دهی وجود دارد:

(۱) ذخیره بخشی از فضای آدرس دهی مجازی که ناحیه یا Region خوانده می شود.

(۲) اعزام کردن یک بلوک از صفحه در یک ناحیه در فضای آدرس دهی.

در مرحله اول یک نخ یا Thread در یک فرآیند بطور پویا یک ناحیه یا Region از فضای

آدرس دهی را بدون اینکه واقعاً موجب شود چیزی در حافظه ثانویه فایل صفحه یا Page File

نوشته شود رزرو یا ذخیره می کند. یک Thread در فرآیند ممکن است متعاقباً یک ناحیه از

آدرس ها که اخیراً ذخیره کرده است را آزاد کند.

در مرحله دوم پردازنده آدرسهای که اخیراً ذخیره شده اند را اعزام می کند ( بلوک اعزام شده غالباً یک زیر مجموعه مناسب از ناحیه ذخیره شده است ).

زمانیکه یک بخش از فضای آدرس دهی اعزام یا فرستاده می شود فضایی در فایل صفحه تخصیص داده می شود. اگر بعداً یک نخ یا thread در فرآیند به حافظه اعزام شده رجوع کند صفحه ای که شامل آدرس رجوع شده است از فایل صفحه به حافظه اولیه بارگذاری می شود. (البته بخشی از فضای آدرس که ذخیره شده است و اعزام شده است در اولین رجوع به آن، چیزی که در آن نوشته شده باشد نخواهد داشت. بنابراین یک صفحه پر شده با صفر در اولین ارجاع بارگذاری خواهد شد.

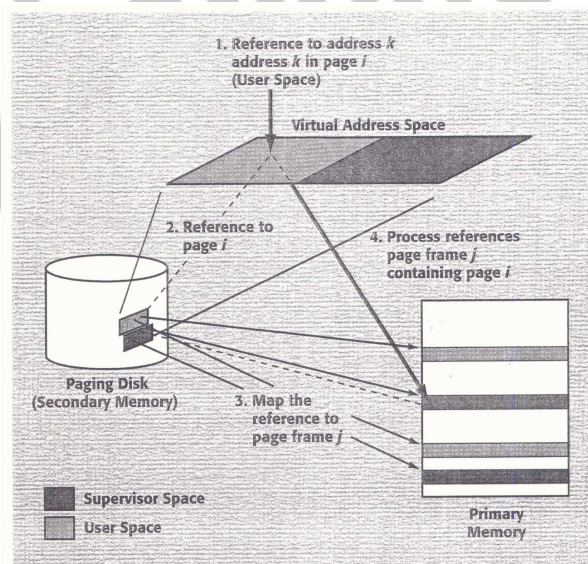
هر پردازنده ای یک دانه بندی تخصیص فضای ویژه ای برای تعیین حداقل اندازه یک بلوک از آدرسها که می توانند ذخیره شوند را پشتیبانی می کند. در همه پیاده سازی های فعلی دانه بندی تخصیص فضا ۶۴k می باشد. (بهر حال شما می توانید برای تعیین هر دانه بندی تخصیص فضای پردازنده از Getsysteminfo استفاده کنید (به جزئیات بعدی در این تمرین نگاه کنید) قبل از اینکه عمل ذخیره سازی صورت گیرد آدرس بصورت خودکار بسمت پایین به مرز دانه بندی تخصیص فضای بعدی پایین تر از خود گرد می شود. هر پردازنده ای همچنین

از اندازه صفحه خود که معمولاً 4K یا 8K است پشتیبانی می کند ( شما می توانید اندازه صفحه پردازنده را با استفاده از GetsystemInfo تعیین کنید ).

حافظه به واحدهایی از صفحه ها سپرده و اعزام می شود بنابراین اعزام واقعی آدرسها می تواند در دانه های کمی کوچکتر از فضای ذخیره سازی اجمام گیرد. زمانی که آدرسهای مجازی اعزام می شوند. نخ یا Thread می تواند از حافظه فقط مانند یک بخش تخصیص فضای ثابت از فضای آدرس دهی استفاده کند.

## داخل سیستم صفحه بندی

ترجمه آدرس به این منظور که عبارت است از تبدیل کردن یک ارجاع آدرس مجازی به



شکل شماره ۲- سیستم صفحه



ارجاع حافظه اولیه فیزیکی اشاره دارد. (مراحل ۱ تا ۳ در شکل ۲)

تبدیل آدرس به حضور عناصر خاصی در سخت افزار برای تعیین کردن صفحات گم شده و انجام سریع منطبق کردن صفحات به صفحه فریم وابسته است. یک آدرس مجازی عبارت است از یک مقدار ۳۲ بیتی که توسط پردازنده تولید شده است. در تقابل با مکانیزم صفحه بندی های مرسوم ویندوز NT یک قابلیت تبدیل آدرس دو سطحی را بکار می برد ( به شکل ۳ نگاه کنید ).

صفحه  $K_1$  Byte index تا از حداقل بیت های مهم در آدرس را در بردارد.  $K_1$  برای پردازنده های intel  $\times 86$  با  $K_4$  اندازه صفحات معادل عدد ۱۲ و برای پردازنده های Digital Alpha با  $8K$  اندازه صفحات معادل عدد ۱۳ می باشد.

مکانیزم های صفحه بندی تک سطحی مرسوم از مقادیر باقیمانده آدرسها بعنوان شماره صفحه استفاده می کنند در ویندوز NT این مقدار باقیمانده شماره صفحه مجازی خوانده می شود که

به ۲ بخش تقسیم می شود که Page table index ( شامل ۲ بیت ) و Page directorty

index ( که شامل  $K_2$  تا از بیشترین بیت های مهم آدرس است ) نام دارند. در پردازنده های

$\times 86$  مقدار  $K_2$  و  $K_3$  برابر ۱۰ است و در پردازنده ها  $K_2$  Alpha برابر ۱۱ و  $k_3$  برابر ۸

می باشد.

تبدیل آدرس از سه فیلد آدرس به شرح زیر استفاده می کند:

(۱) توصیف کننده فرآیند یا همان Process descriptor شامل یک اشاره گر مثل A می باشد

که به ابتدای Page directory برای فرآیند داده شده اشاره می کند.

(۲) Page directory index که a است یک offset در Page directory است جایی که یکم

Page descriptor entry یا ( PDE ) برای آن صفحه خاص در آن واقع شده است.

(۳) هر فرآیندی می تواند چندین جدول صفحه ( Page Table ) مختلف داشته باشد. PDE به

جدول صفحه مجزایی که برای ارجاعات حافظه استفاده می شود رجوع می کند. ( اشاره گر B در شکل ).

(۴) Page Table entry ( PTE ) با استفاده از Page table index که b است از آدرس که

بعنوان یک اندیس به Page Table است یافته می شود.

(۵) اگر صفحه مقصد یا target page در حال حاضر در فریم صفحه در حافظه اولیه بارگذاری

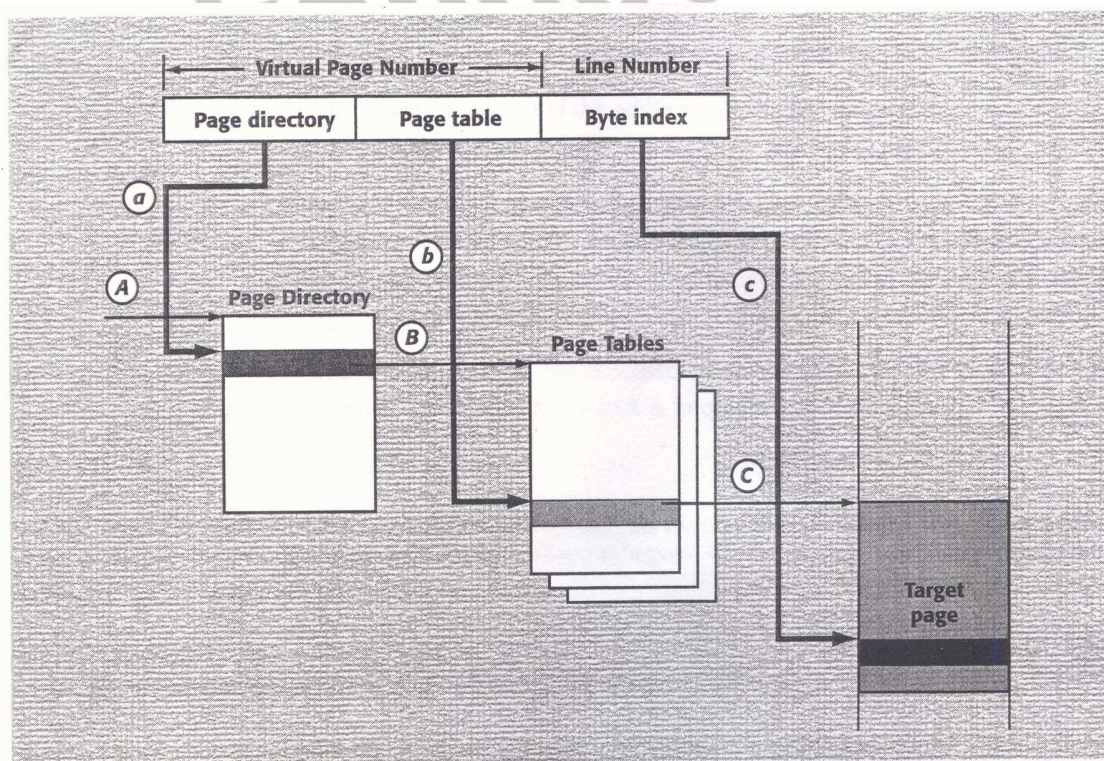
شده بود در آن صورت PTE به فریم صفحه رجوع می کند که C در شکل است. اگر بارگذاری

نشده بود مدیریت حافظه مجازی باید موقعیت صفحه را در فایل صفحه پیدا کند و بعد فریم

صفحه را پیدا کرده و آن را به فرآیند اختصاص دهد و بعد صفحه را در فریم صفحه بارگذاری

کند.

۶) در آخر byte index (C) به آدرس مبنای فریم صفحه اضافه می شود برای اینکه موقعیت  
بایت مقصد در حافظه اولیه بدست آید.



شکل شماره ۳- تبدیل آدرس



تا یک زمانی Page directory می تواند به هر مکانی اشاره کند (اشاره گر A در شکل ۳)

در تمرین در یک نقطه ثابت در فضای آدرس دهی که  $0 \times C0300000$  در سیستمهای X86 است و  $0 \times C018000$  در پردازنده های APPLE است واقع شده است. هر دو پردازنده از ثباتهای اختصاصی داده شده به آنها برای رجوع به Page directory استفاده می کنند ثبات بعنوان بخشی از محتوای Thread ذخیره می شود زمانی که یک Thread جدید به پردازنده داده می شود. ویندوز NT از چندین جدول صفحه (Page Table) برای تشخیص فضای آدرس دهی در میان کاربردهای مختلف استفاده می کند. بیشترین اختلاف مشهود در میان صفحه ها این است که برخی، صفحات فضای کابر هستند و دیگری ها صفحات فضای هسته (Kernel) که بر جداول صفحه مختلفی منطبق می شوند. توجه کنید چونکه صفحات فضای Kernel در جدول صفحه مجزای گذاشته می شود فرآیندهای مختلف، PDE هایی برای فضای Kernel دارند که به جداول صفحه یکسانی برای بخشهای مجزایی از فضای آدرس دهی بکار برده می شود را برای مثال در فضای Kernel چند قسمت از حافظه Kernel درست مطابق آنچه در صفحه بندی فضای user است بسمت داخل و خارج صفحه بندی شده است



اما دیگران که از بولکهای حافظه اجازه ندارد بسمت خارج صفحه بندی کنند تقریباً آنها از یک استخر ویژه ای که از بلوکهای حافظه غیر صفحه بندی شده تشکیل شده است تعیین می شوند. هر PTE وقتی صفحه مرتبط با آن بارگذاری شد به یک شماره فریم صفحه اشاره می کند همچنین یک مجموعه از Flag ها برای شرح اینکه چگونه صفحه می تواند مورد ارجاع قرار گیرد وجود دارد که شامل این است که آیا PTE قانونی است؟ و آیا صفحه ذخیره شده است؟ یا کثیف است؟ ( یعنی در صفحه مقدار جدیدی از زمانی که آخرین بار در حافظه اولیه بارگذاری شده نوشته شده است یا به ) یا به آن از زمان آخرین دوره ایجاد آن دسترسی پیدا شده است یا به واز قبیل اینها.

رجوع به حافظه نیاز دارد که PDE در Page directory یافته شود و همچنین PTE هم در جدول صفحه یا Page Table یافته شود این بدان معنی است که یک رجوع حافظه معمولی می تواند نتیجه رجوعات حافظه های اضافه مختلفی باشد چنانچه سخت افزار ویژه در پردازنده تأمین شده باشد. کامپیوترهای جاری مانند پردازنده های intel  $\times 86$  و Digital Alpha از حافظه های انجمنی که Translation look – asid Buffer ( TLB ) خوانده می شوند استفاده می کنند. بطور خلاصه TLB یک نوع حافظه Cache برای ترجمه های اخیر PDE ها و PTE ها است.

اولین رجوع به چنین مدخل‌هایی موجب رجوعات حافظه زیاد برای دسترسی به مدخل‌ها می‌شود اما زمانی که یک صفحه map می‌شود تمام مدخل‌های آن در TLB کپی می‌شود. رجوعات بعدی بصورت انجمنی در TLB جستجو می‌شوند مکانی که آنها می‌توانند در یک TLB واحد در یک دوره حافظه انجمنی یافته شوند.

ویندوز NT از صفحه‌بندی درخواستی یا Demand Paging استفاده می‌کند به این معنا که صفحات در حافظه اولیه بارگذاری نمی‌شوند تا وقتی که به آنها رجوع شود بعلاوه PTE ها هم حتی ایجاد نشده‌اند تا زمانی که صفحه مربوطه بارگذاری شود منطق این دستاورد این است که فرآیند ممکن است آدرسی از حافظه را اشغال کند و بدون اینکه از آن استفاده کند در واقع فرآیند ممکن است حتی صفحات حافظه را اعزام کند و بعد هرگز در حین اجرا به آنها رجوع نکند. زمانی که فضای آدرس‌دهی خیلی بزرگ است اگر PTE ها در همان زمانی که اطلاعات در حال شناخته شدن است ساخته شوند تعداد زیادی PTE بصورت مشابه ممکن است ایجاد شوند و بعد هرگز استفاده نشوند که موجب ایجاد حافظه هدررفته باورنکردنی برای PTE ها می‌شوند. بدلیل اینکه PTE ها ایجاد نمی‌شوند تا اینکه یکبار از آنها استفاده شود سیستم عامل باید دیگر ساختارهای داده‌ای و همینطور عملیات اعزام یا فرستادن رادر حالت ذخیره نگه دارد. توصیف کننده آدرس مجازی یا Virtual address descriptor (VAD) زمانی که یک

فرآیند ذخیره می شود یا آدرس مجازی را اعزام می کند و می فرستد ایجاد می شود. زمانی که یک Thread برای بار اول به یک آدرس در میان VAD رجوع می کند PTE سپس ایجاد می شود بنابراین ترجمه آدرس می تواند بصورت طبیعی پیش برود.

آخرین جنبه داخلی صفحه بندی که در اینجا به آن اشاره می شود تخصیص دهی حافظه اولیه می باشد ویندوز NT از مجموعه های کاری با یک الگوریتم زمان بندی شده استفاده می کند. به این صورت بین مجموعه کاری فرآیند و مجموعه کاری سیستم که توسط فرآیند مورد استفاده قرار می گیرد وجه تمایز قائل می شود. مجموعه کاری فرآیند پیش می رود و کوچک می شود مانند هر مجموعه کاری کلاسیک و با حداقل اندازه پیش فرض که بین ۲۰ تا ۵۰ صفحه می باشد شروع به کار می کند و اجازه ندارد بیشتر از حداکثر اندازه پیش فرض که ۴۵ تا ۳۴۵ صفحه است رشد کند. بهر حال حداکثر اندازه مجموعه کاری می تواند بوسیله مدیر سیستم تغییر پیدا کند. جزئیات زیادی در مورد سیستم حافظه مجازی ویندوز NT وجود دارد و این جزئیات می تواند بسته به نوع پردازنده قرار گرفته شده تغییر یابد. این زیر بخش شما را با معرفی عملیات مدیریت حافظه مجازی آگاه می کند برای کسب مباحث کاملتر در مورد مکانیزم صفحه بندی ویندوز NT به بخش ۵ از کتاب Solomon ( ۱۹۹۸ ) مراجعه کنید.

## توابع حافظه مجازی

واسطه‌های برنامه‌نویسی کاربردی Win32 توابع زیادی برای رسیدگی به وضعیت حافظه مجازی

و کنترل پویای راههای بکارگیری فضای آدرس دهی فراهم می‌کند.

Virtual Query , GlobalMemorystatus, GetSystemInfo می‌توانند برای رسیدگی به

وضعیت حافظه مجازی یک فرآیند مورد استفاده قرار بگیرند اطلاعات پایه‌ای بیشتر در این

مورد در زیر آورده شده است.