

فصل هشتم

حافظه مجازی

وحید رنجبر

پائیز ۹۸

مباحث این فصل:

■ ساختارهای سخت افزاری و کنترل

- محلی بودن و حافظه مجازی
- صفحه بندی
- قطعه بندی
- ترکیب صفحه بندی و قطعه بندی
- حفاظت و اشتراک

■ نرم افزار سیستم عامل

- سیاست واکشی
- سیاست جاگذاری
- سیاست جایگزینی
- مدیریت مجموعه مقیم
- سیاست پاکسازی
- کنترل بار

ساختار های سخت افزاری و کنترل:

■ دو مشخصه اصلی صفحه بندی و قطعه بندی کلید پیشرفت مدیریت حافظه اند:

■ کلیه مراجعات یک فرایند به حافظه، آدرسهای منطقی هستند که به صورت پویا در زمان اجرا به آدرس فیزیکی ترجمه میشوند.

■ فرایند ممکن است به داخل و خارج حافظه مبادله شود بنابراین ممکن است بخشهای مختلفی از حافظه را اشغال کند.

■ ممکن است برنامه به بخش های تقسیم شود که نیاز نیست به صورت پیوسته در حافظه قرار گیرند.

■ برای اجرای فرایند نیاز به بار شدن تمام بخشهای فرایند نیست.

■ با ترکیب ترجمه آدرس پویا در زمان اجرا و به کارگیری جدول صفحه این کار امکان پذیر است.

حافظه مجازی و اجرای برنامه:

- سیستم عامل بخشی از فرایند را به حافظه اصلی بار میکند.
- مجموعه مقیم: بخشی از فرایند که در حافظه است.
- زمانی که پردازنده به آدرس منطقی نیاز دارد که در حافظه وجود ندارد، وقفه‌ای به معنای خطای دسترسی به حافظه تولید میکند.
- سیستم عامل فرایند وقفه خورده را مسدود میکند و کنترل را بدست میگیرد.

حافظه مجازی و اجرای برنامه:

- سیستم عامل آن قسمت از فرایند که تولید کننده خطای حافظه است را را بداخل بار میکند.
- سیستم عامل یک درخواست خواندن به دیسک صادر میکند.
- در حین عمل I/O سیستم عامل فرایند دیگری را اجرا میکند.
- زمانی که عمل I/O کامل شد، یک وقفه ورودی/خروجی صادر میشود و موجب میشود فرایند مسدود به حالت آماده تغییر حالت دهد.

مزایای تقسیم فرایند:

- فرایند های بیشتری میتوانند در حافظه نگه داشته شوند.
- تنها بخشی از فرایند به حافظه بار میشود.
- با داشتن فرایندهای بیشتر در حافظه احتمال وجود فرایند آماده به اجرا بیشتر میشود و این موجب افزایش کارایی پردازنده میشود.
- یک فرایند میتواند از حافظه اصلی بزرگتر باشد.

انواع حافظه:

- حافظه حقیقی:

- حافظه اصلی

- حافظه مجازی:

- یک حافظه بالقوه خیلی بزرگتر از حافظه اصلی

- حافظه روی دیسک

- چند برنامه‌گی را بصورت مؤثرتری ممکن می‌سازد و کاربر را از محدودیت‌های حافظه اصلی رها میکند.

جدول ۸-۱: مشخصات صفحه‌بندی و قطعه‌بندی

صفحه‌بندی ساده	حافظه مجازی با صفحه‌بندی	قطعه‌بندی ساده	حافظه مجازی با قطعه‌بندی
حافظه اصلی به تکه‌های هم‌اندازه به نام قاب تقسیم می‌شود.	حافظه اصلی به تکه‌های هم‌اندازه به نام قاب تقسیم می‌شود.	حافظه اصلی تقسیم نمی‌شود.	حافظه اصلی تقسیم نمی‌شود.
برنامه توسط مترجم یا سیستم مدیریت حافظه به صفحه‌ها تقسیم می‌شود.	برنامه توسط مترجم یا سیستم مدیریت حافظه به صفحه‌ها تقسیم می‌شود.	قطعه‌های برنامه توسط برنامه‌ساز به مترجم اطلاع داده می‌شوند (یعنی تصمیم‌گیری با برنامه‌ساز است).	قطعه‌های برنامه توسط برنامه‌ساز به مترجم اطلاع داده می‌شوند (یعنی تصمیم‌گیری با برنامه‌ساز است).
تکه‌تکه شدن داخلی درون قاب	تکه‌تکه شدن داخلی درون قاب	بدون تکه‌تکه شدن داخلی	بدون تکه‌تکه شدن داخلی
بدون تکه‌تکه شدن خارجی	بدون تکه‌تکه شدن خارجی	تکه‌تکه شدن خارجی	تکه‌تکه شدن خارجی
سیستم عامل بایستی یک جدول صفحه برای هر فرایند نگهداری کند تا نشان دهد هر صفحه در کدام قاب است.	سیستم عامل بایستی یک جدول صفحه برای هر فرایند نگهداری کند تا نشان دهد هر صفحه در کدام قاب است.	سیستم عامل بایستی یک جدول قطعه برای هر فرایند نگهداری کند و آدرس بار شدن و طول هر قطعه را نشان دهد.	سیستم عامل بایستی یک جدول قطعه برای هر فرایند نگهداری کند و آدرس بار شدن و طول هر قطعه را نشان دهد.
سیستم عامل باید لیست قابهای آزاد را نگهداری کند.	سیستم عامل باید فهرست قابهای آزاد را نگهداری کند.	سیستم عامل باید فهرست حفره‌های آزاد در حافظه اصلی را نگهداری کند.	سیستم عامل باید فهرست حفره‌های آزاد در حافظه اصلی را نگهداری کند.
پردازنده از شماره صفحه و انحراف برای محاسبه آدرس مطلق استفاده می‌کند.	پردازنده از شماره صفحه و انحراف برای محاسبه آدرس مطلق استفاده می‌کند.	پردازنده از شماره قطعه و انحراف برای محاسبه آدرس مطلق استفاده می‌کند.	پردازنده از شماره قطعه و انحراف برای محاسبه آدرس مطلق استفاده می‌کند.
تمام صفحه‌های یک فرایند باید در حافظه اصلی باشند تا فرایند اجرا شود، مگر اینکه روی هم‌گذاری به کار رفته باشد.	تمام صفحه‌های یک فرایند باید در حافظه اصلی باشند تا فرایند اجرا شود، مگر اینکه روی هم‌گذاری به کار رفته باشد.	تمام قطعه‌های یک فرایند باید در حافظه اصلی باشند تا فرایند اجرا شود، مگر اینکه روی هم‌گذاری به کار رفته باشد.	تمام قطعه‌های یک فرایند باید در حافظه اصلی باشند تا فرایند اجرا شود، مگر اینکه روی هم‌گذاری به کار رفته باشد.
خواندن یک صفحه به داخل حافظه اصلی، ممکن است نیازمند نوشتن یک صفحه بر روی دیسک باشد.	خواندن یک صفحه به داخل حافظه اصلی، ممکن است نیازمند نوشتن یک صفحه بر روی دیسک باشد.	خواندن یک قطعه به داخل حافظه اصلی، ممکن است نیازمند نوشتن یک قطعه بر روی دیسک باشد.	خواندن یک قطعه به داخل حافظه اصلی، ممکن است نیازمند نوشتن یک قطعه بر روی دیسک باشد.

کوبیدگی:

- اگر سیستم عامل تکه ای را دقیقاً قبل از اینکه به کار گرفته شود خارج کند، با فاصله کمی باید آنرا به داخل بیاورد.
- با اجرای زیاد این عمل پردازنده بیشتر وقت خود به جای اجرای برنامه صرف مبادله تکه ها می کند، به این عمل **کوبیدگی** میگویند.

اصل محلیت:

- در یک فرایند مراجعات به برنامه و داده ها در حافظه خوشه ای هستند.
- در یک بازه زمانی کوتاه فقط به تعداد کمی از تکه های یک فرایند نیاز است.
- باید حدسهای هوشمندانه ای زد که به کدام تکه های یک فرایند در آینده نزدیک نیاز است.
- پیشنهاد میکند که طرح حافظه مجازی میتواند کار کند.

حمایت های لازم برای حافظه مجازی

- سخت افزار باید از صفحه بندی و قطعه بندی حمایت کند.
- سیستم عامل باید قادر به مدیریت انتقال صفحه ها و یا قطعه ها بین حافظه ثانویه و حافظه مجازی باشد.

صفحه بندی:

- هر فرایند جدول صفحه مخصوص به خود دارد.
- هر ورودی مدخل صفحه شامل شماره قاب متناظر با صفحه داخل حافظه اصلی است.
- بیت حضور وجود صفحه در حافظه اصلی را نشان میدهد.

صفحه بندی:

Virtual Address



Page Table Entry

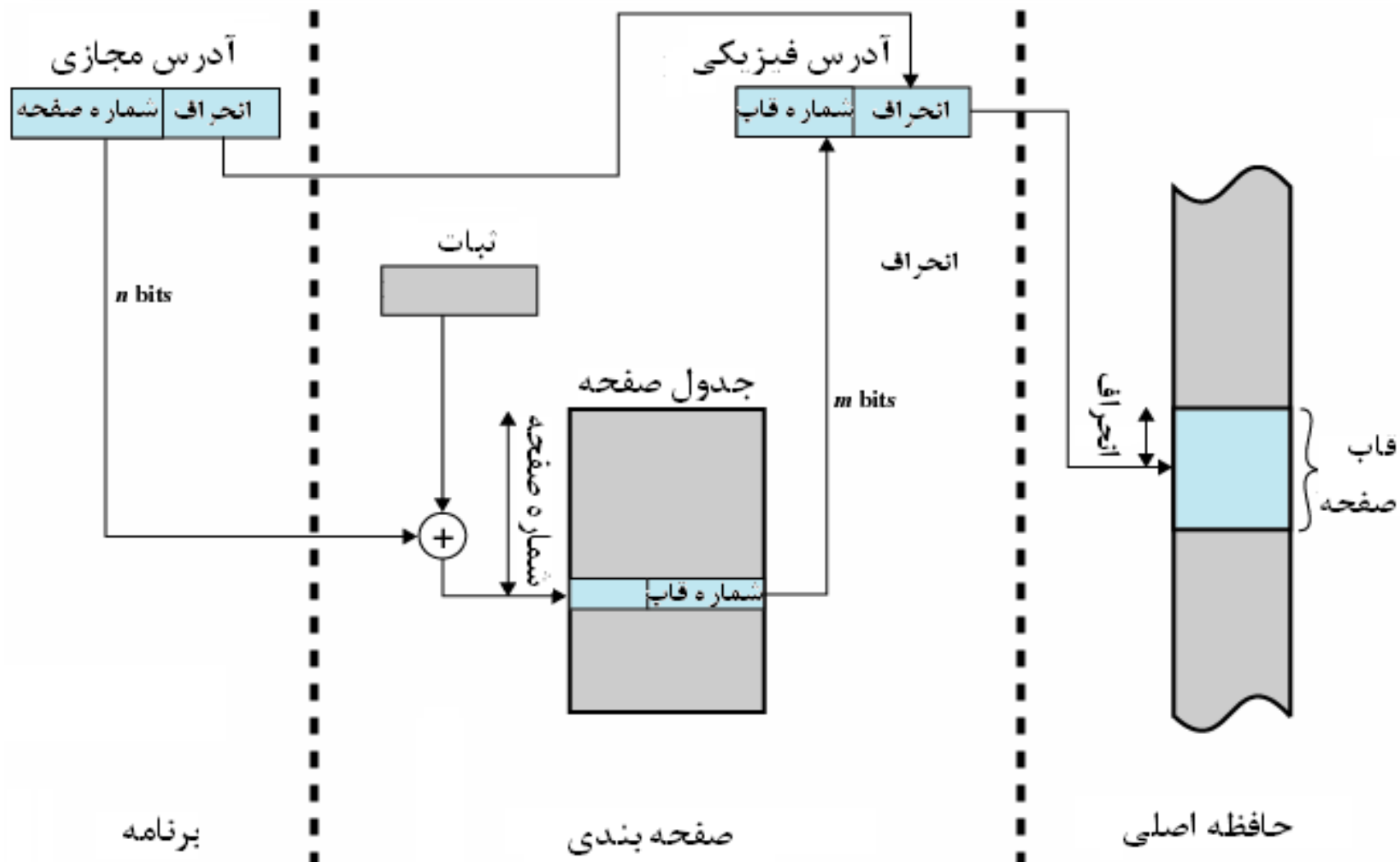


(a) Paging only

بیت اصلاح در جدول صفحه:

- این بیت مبین تغییر محتوای یک صفحه پس از آخرین بار شدن در حافظه اصلی است.
- اگر هیچ تغییری در صفحه انجام نشده باشد، نیازی به نوشتن صفحه روی دیسک هنگام تعویض آن صفحه نیست.

ترجمه آدرس در یک سیستم صفحه بندی:



جدول صفحه دو سطحی:

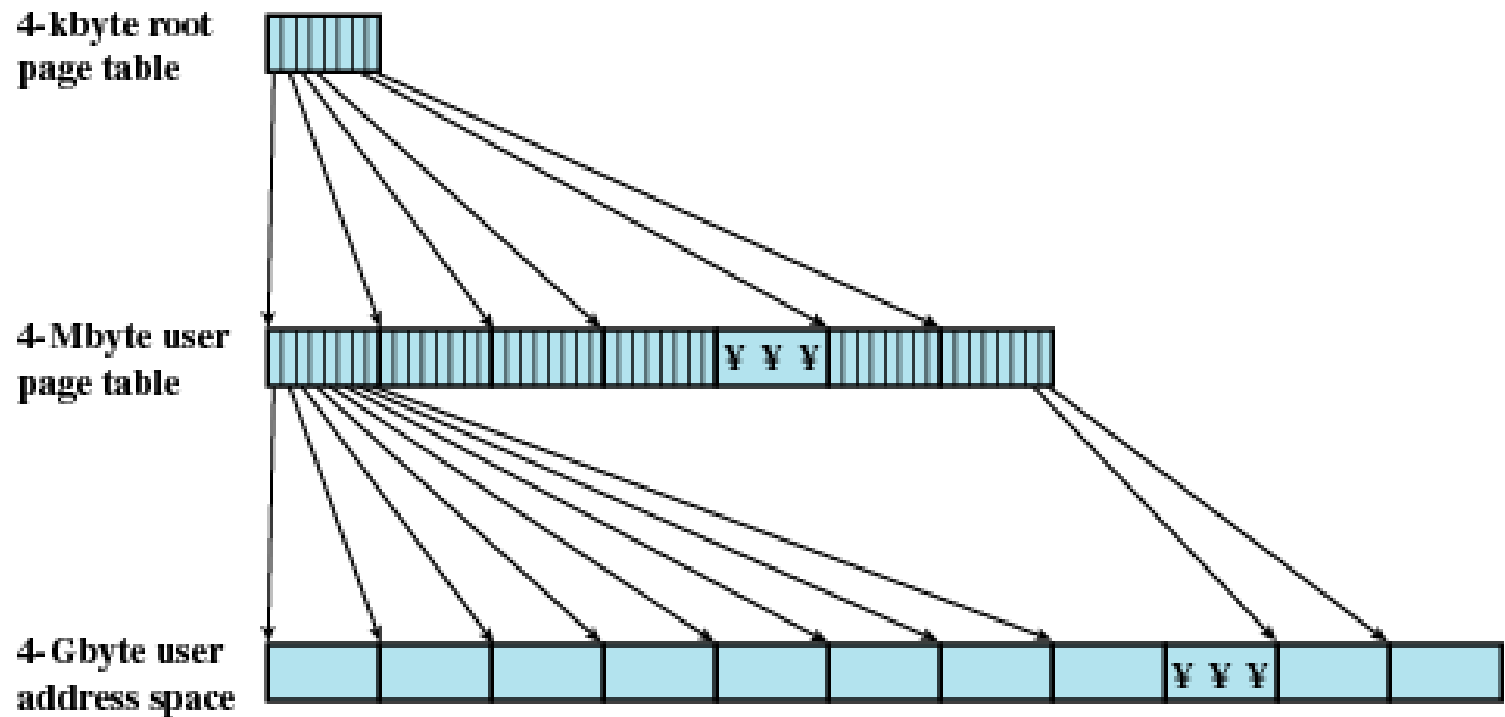


Figure 8.4 A Two-Level Hierarchical Page Table

جدول صفحه:

- مدخل جدول صفحه میتواند مقدار زیادی از حافظه اصلی را اشغال کند.
- جدول صفحه نیز میتواند در حافظه مجازی ذخیره شود.
- هنگامی که یک فرایند اجرا میشود بخشی از جدول صفحه آن در حافظه اصلی قرار دارد.

جدول صفحه معکوس:

- قسمت شماره صفحه از آدرس مجازی با استفاده از یک تابع درهم ساز ساده به یک جدول درهم نگاشته میشود.
- جدول درهم دارای اشاره گری به جدول معکوس است که شامل مدخل های جدول صفحه است.
- بدون توجه به تعداد فرایندها و یا صفحه های مجازی بخش ثابتی از حافظه حقیقی برای جدولها مورد نیاز است.

جدول صفحه معکوس:

■ جدول صفحه معکوس شامل موارد زیر است:

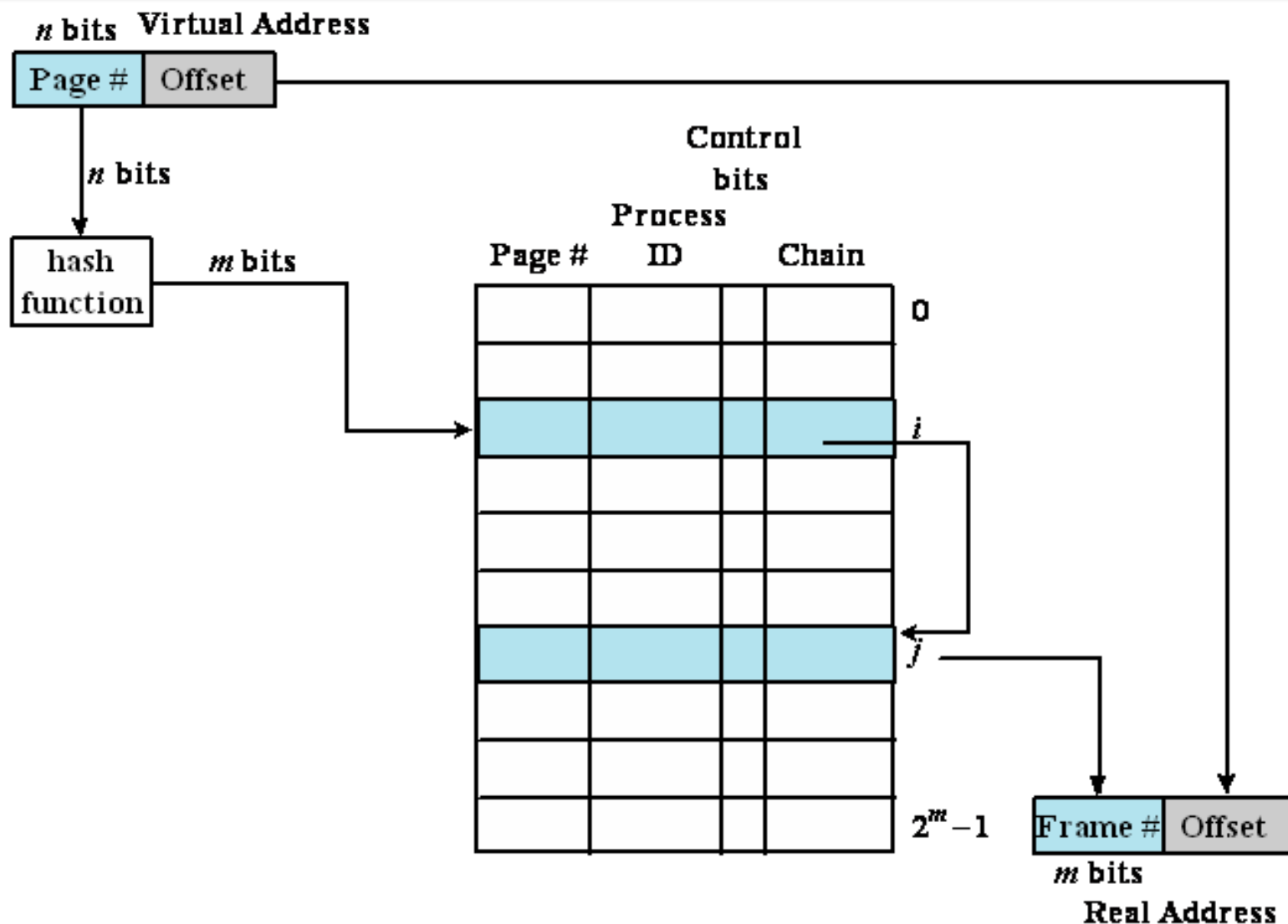
■ شماره صفحه

■ بیت‌های کنترلی

■ شناسه فرایند

■ اشاره گر زنجیره

ساختار جدول صفحه معکوس:



میانگیر دم دستی ترجمه:

- هر ارجاع به حافظه مجازی موجب دو دسترسی به حافظه فیزیکی را میشود:
- یکی برای واکشی از مدخل جدول صفحه مربوط
- یکی برای واکشی داده مورد نظر
- برای غلبه بر این مشکل یک حافظه بسیار سریع نهان برای مدخل های جدول صفحه استفاده میکنند.
- به این حافظه نهان میانگیر دم دستی گفته میشود.
- حاوی مدخلهایی از جدول صفحه است که اخیراً به آنها دسترسی شده است.

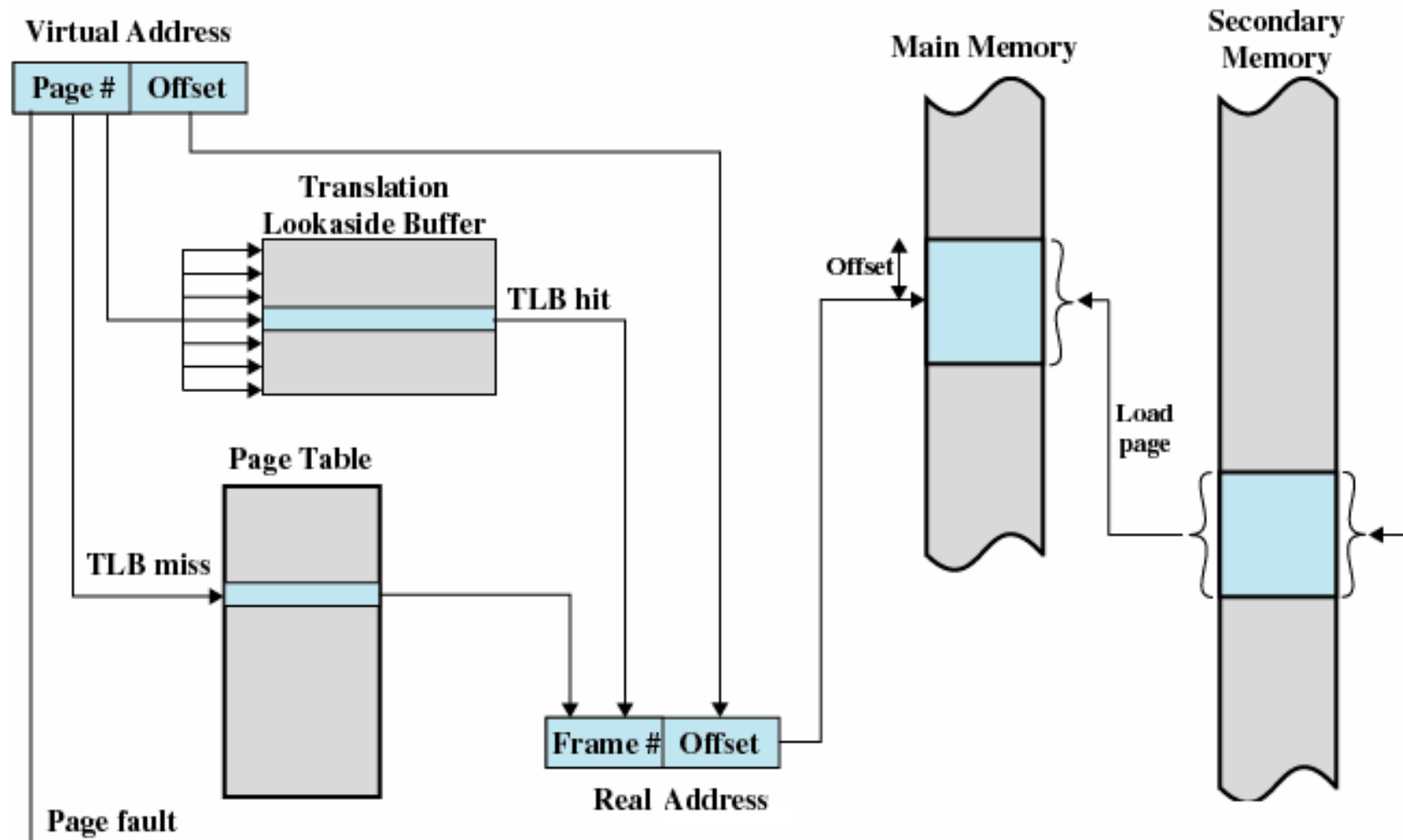
میانگیر دم دستی ترجمه:

- با داشتن یک آدرس مجازی پردازنده میانگیر دم دستی را امتحان میکند.
- اگر مدخل مورد نظر جدول صفحه پیدا شد (اصابت)، شماره قاب بازیابی شده و آدرس حقیقی شکل میگیرد.
- اگر مدخل مورد نظر از جدول صفحه پیدا نشد (عدم اصابت) پردازنده از شماره صفحه به عنوان شاخص جدول صفحه فرایند استفاده میکند و مدخل مورد نظر در جدول صفحه را آزمایش میکند.
- بیت حضور (حضور صفحه در حافظه) بررسی میشود:
 - اگر در حافظه اصلی نبود خطای **فقدان صفحه** (Page Fault) صادر میشود.
 - اگر در حافظه اصلی بود، پردازنده شماره قاب را از مدخل جدول صفحه برای تشکیل آدرس حقیقی بدست می آورد و میانگیر دم دستی را بهنگام میکند تا شامل این مدخل جدید صفحه شود.

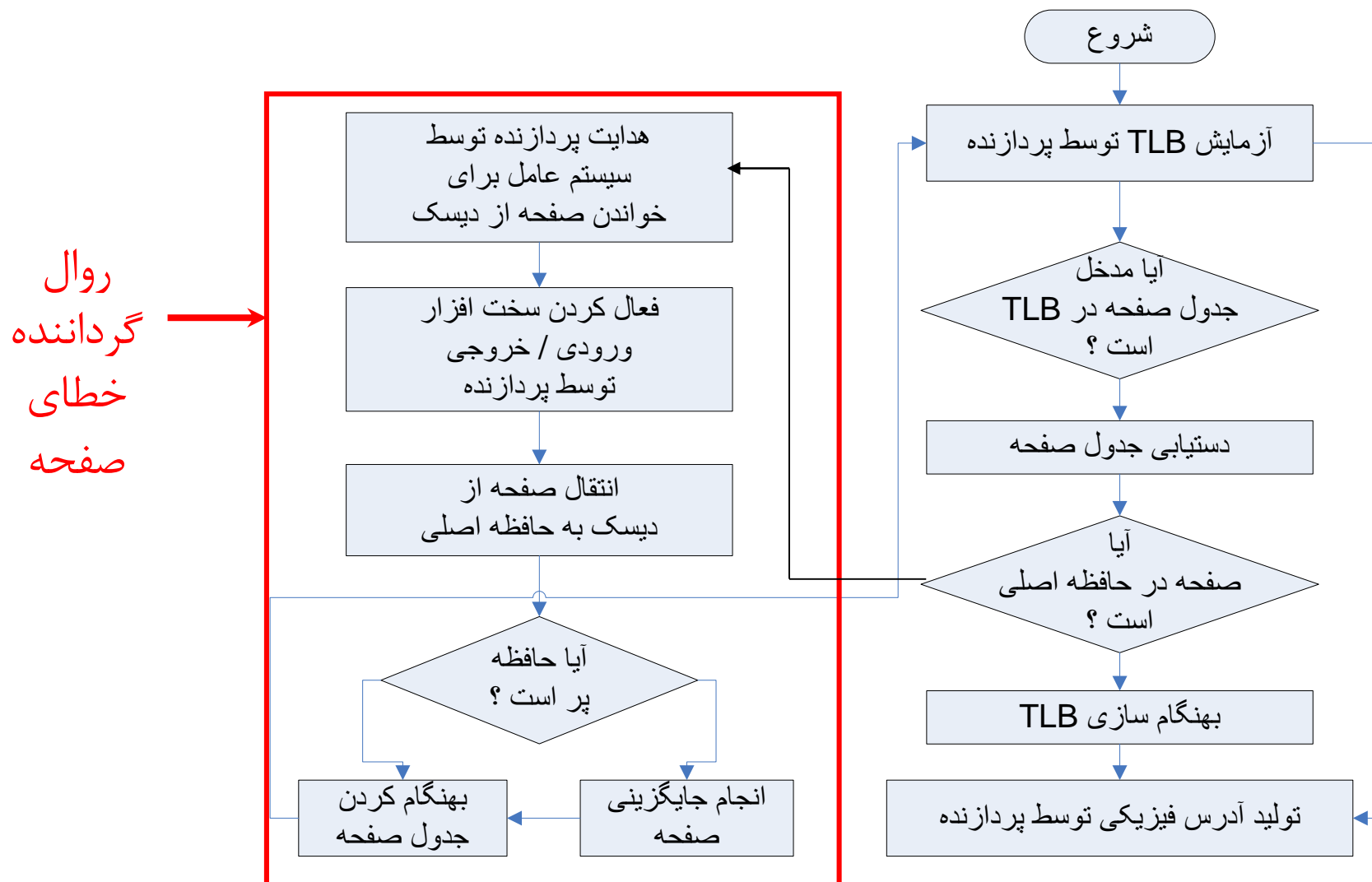
میانگیر دم دستی ترجمه:

- به طور کلی هدف میانگیر دم دستی ترجمه، کم کردن تعداد مراجعه به دیسک برای واکشی مدخل جدول صفحه است.
- با توجه به اصل محلی بودن، اغلب مذاجعات به حافظه در صفحه‌هایی که اخیرا به آنها دسترسی شده است، قرار دارد.
- پردازنده مجهز به سخت‌افزاری است که به او اجازه می‌دهد به طور همزمان تعدادی از مدخل‌های میانگیر دم‌دستی را برای یک شماره صفحه واریسی کند. این روش به عنوان **نگاشت انجمنی** شناخته می‌شود و در مقابل روش نگاشت مستقیم یا شاخص‌بندی قرار دارد.

استفاده از میانگیر دم دستی ترجمه:



استفاده از میانگیر دم دستی ترجمه:



میانگیر دم دستی ترجمه و حافظه پنهان

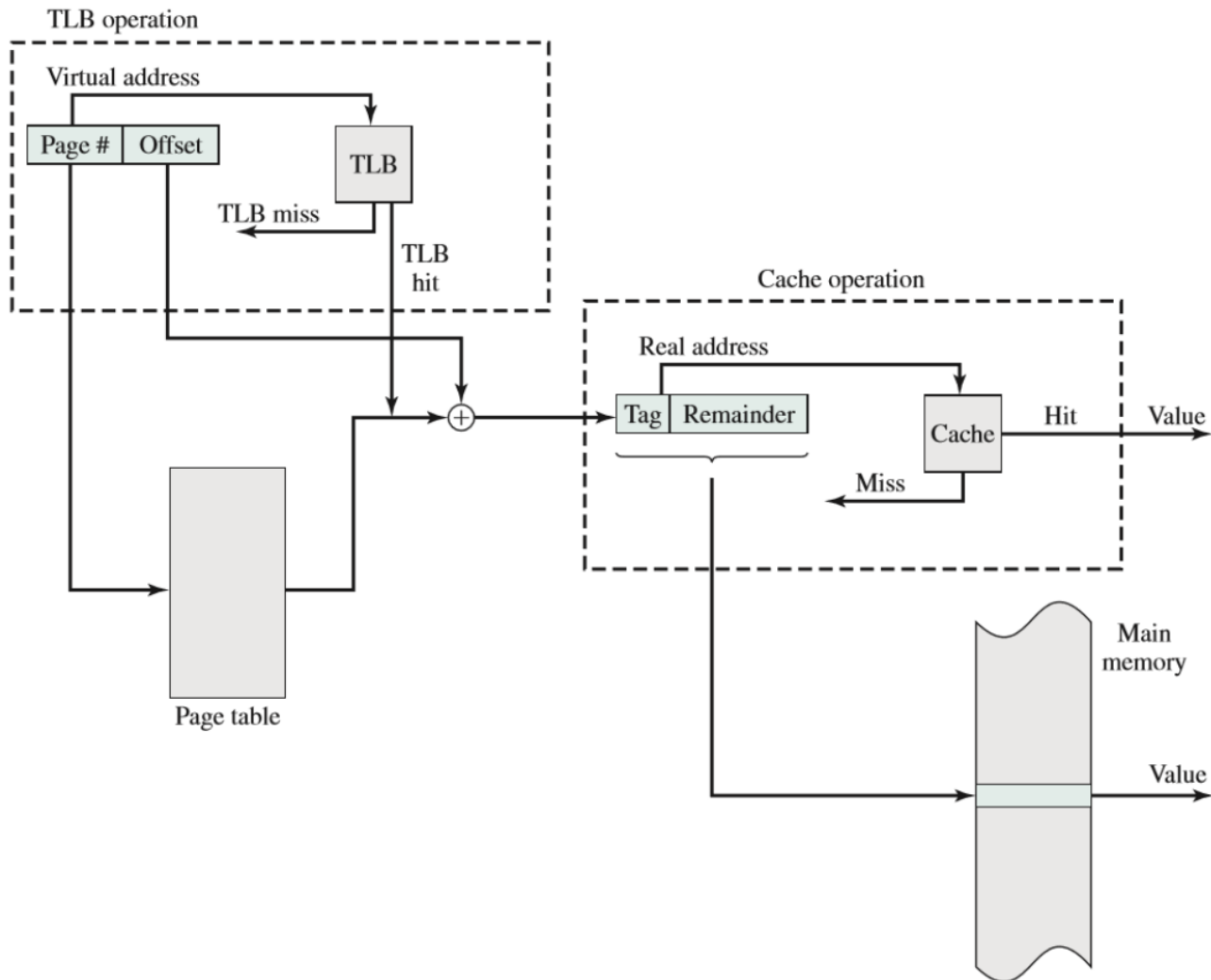
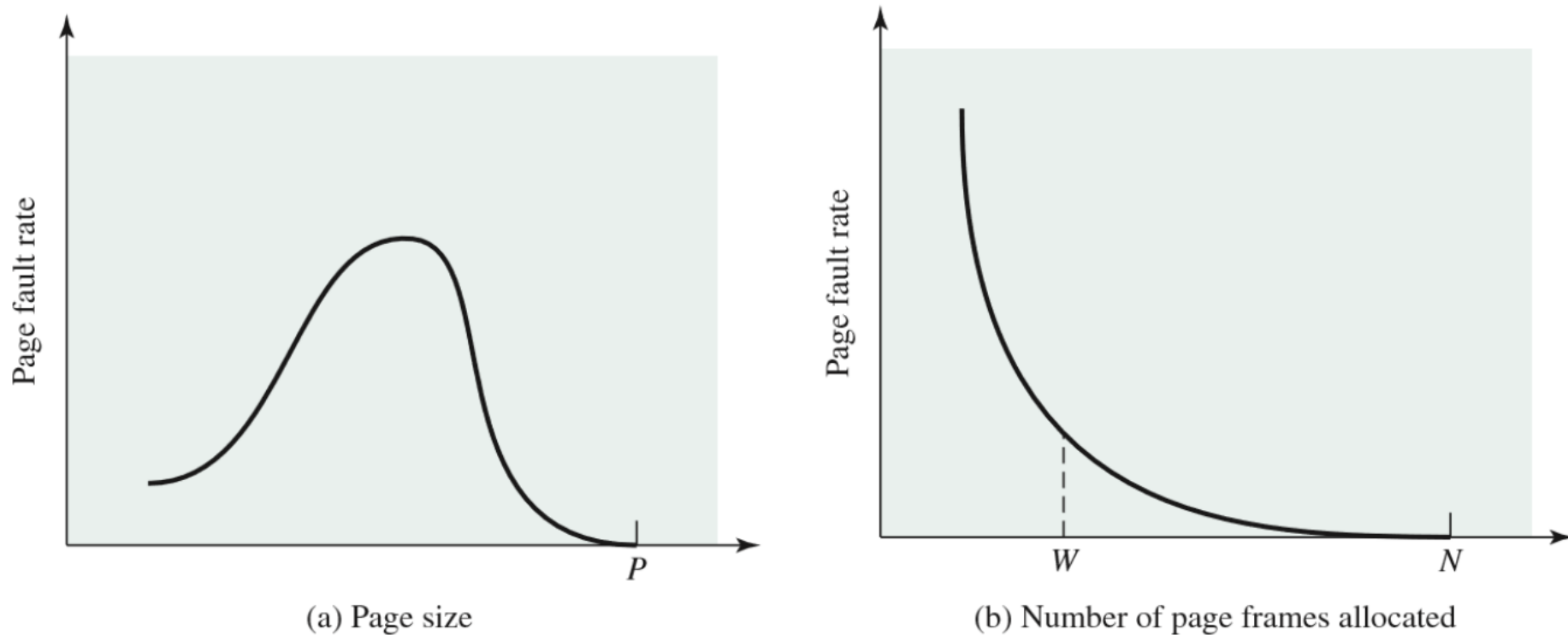


Figure 8.9 Translation Lookaside Buffer and Cache Operation

اندازه صفحه:

- یکی از تصمیمات مهم در طراحی سخت افزار تعیین اندازه صفحه است.
- واضح است که هر چه اندازه صفحه کوچکتر باشد مقدار تکه تکه شدن داخلی آن کمتر می شود.
- از طرفی کوچک شدن اندازه صفحه باعث می شود تعداد صفحه های هر فرآیند افزایش یابد که نیاز به جدول صفحه بزرگتر است.
- مشخصه فیزیکی اکثر دستگاه های حافظه ثانویه، به گونه ای است که برای انتقال کارآمدتر بلوک های داده ها، اندازه صفحه بزرگ مناسبتر است.

رفتار صفحه بندی در یک برنامه



P = size of entire process

W = working set size

N = total number of pages in process

Figure 8.10 Typical Paging Behavior of a Program

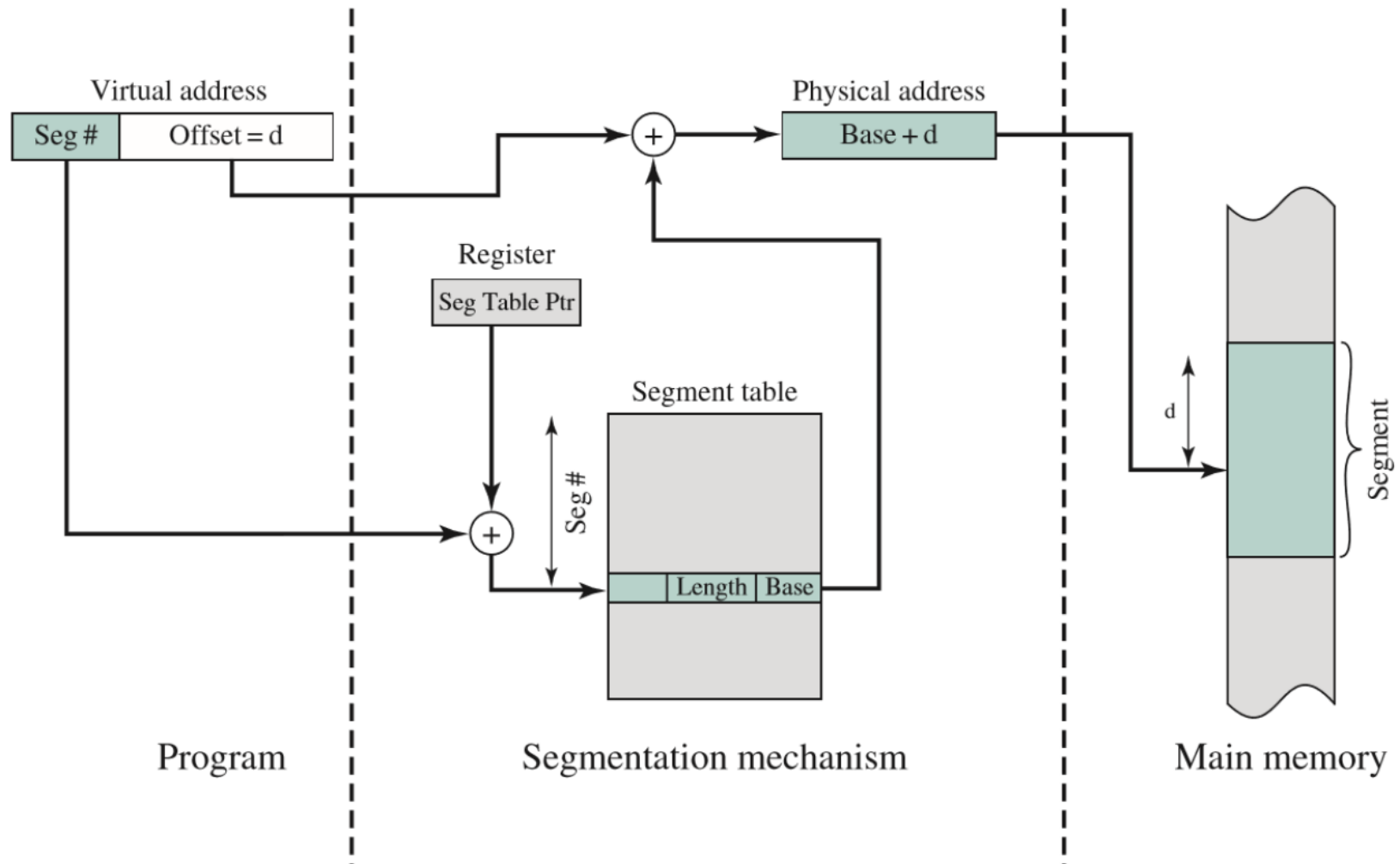
اندازه صفحه:

- اندازه صفحه بر روی نرخ خطای صفحه تاثیر دارد (سخت افزار)
- تعداد قابهای تخصیص یافته به یک فرآیند در نرخ خطای صفحه تاثیر دارد (سیاست نرم افزاری)
- اندازه صفحه به اندازه فیزیکی حافظه اصلی و اندازه برنامه نیز بستگی دارد.
- روش‌های برنامه نویسی جدید تمایل به کاهش محلی بودن دارند.
 - شی گرای
 - چند نخی
- اندازه میانگیر دم دستی ترجمه؟؟؟
 - اندازه‌های متفاوت صفحه

قطعه بندی در حافظه مجازی:

- به برنامه ساز اجازه می دهد که حافظه را به صورت فضای آدرسی چندگانه یا قطعه ها ببیند.
- قطعه ها می توانند در اندازه های نامساوی و واقعا پویا باشند.
- مراجعات به حافظه شامل آدرسهایی به صورت شماره قطعه و انحراف هستند
- مزایای قطعه بندی:
 - اداره ساختمان داده های رشد یابنده را ساده می کند.
 - اجازه می دهد بخش های برنامه مستقلا تغییر یابند.
 - اشتراک بین فرآیندها را میسر می سازد.
 - حفاظت را میسر می سازد.

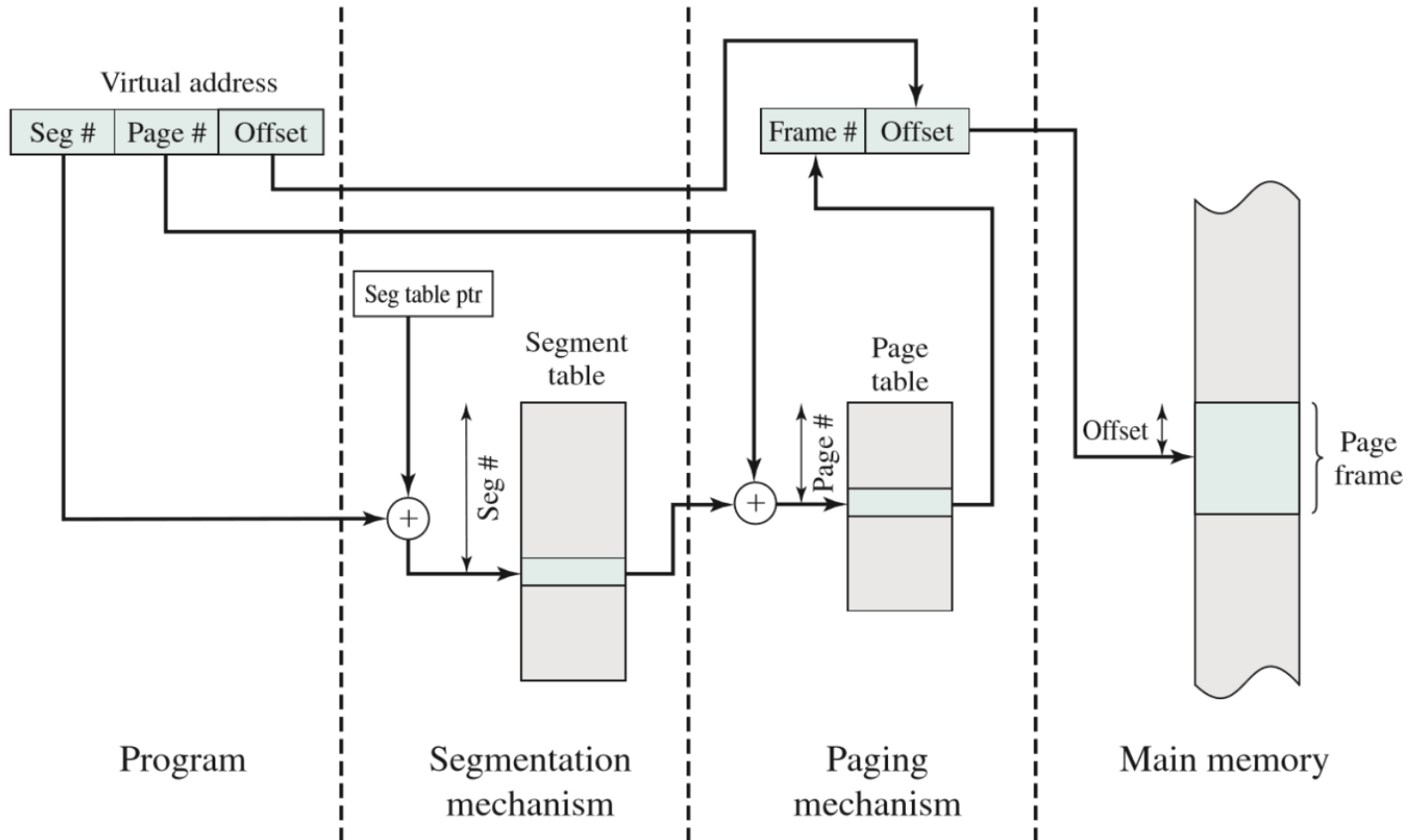
ترجمه آدرس در یک سیستم قطعه بندی:



ترکیب صفحه بندی و قطعه بندی

- فضای آدرس کاربر تحت نظر برنامه ساز به تعدادی قطعه شکسته می شود.
- هر قطعه نیز به نوبه خود به تعدادی صفحه با اندازه ثابت که برابر با اندازه قاب در حافظه است تقسیم می گردد.
- اگر طول یک قطعه کوچکتر از اندازه یک صفحه باشد، آن قطعه یک صفحه را اشغال می کند.
- یک جدول قطعه و چند جدول صفحه (یک جدول صفحه برای هر قطعه)

ترجمه آدرس در یک سیستم صفحه‌بندی / قطعه بندی:



سیاست واگذاری:

- زمان آوردن صفحه به حافظه اصلی را مشخص میکند.
- صفحه بندی درخواستی فقط زمانی یک صفحه را به حافظه اصلی می آورد که مراجعه ای به مکانی از آن صفحه انجام گیرد.
- هنگام شروع برنامه خطاهای صفحه زیادی رخ میدهد.
- در پیش صفحه بندی، صفحه هایی غیر از آنچه به وسیله خطای صفحه درخواست شده نیز به داخل آورده میشوند.

سیاست جایگذاری:

- محل قرار گرفتن فرایند در حافظه اصلی را تعیین میکند.
- در یک سیستم قطعه بندی ساده سیاست جایگذاری بسیار مهم است.
- سیاست جایگذاری در سیستم صفحه بندی یا ترکیب صفحه بندی و قطعه بندی مهم نیست. زیرا سخت افزار ترجمه آدرس را انجام میدهد.

سیاست جایگزینی:

- سیاست جابجایی شامل موارد زیر است:
 - کدام صفحه جایگزین شود؟
 - صفحه حذف شده باید کمترین مراجعه در آینده نزدیک را داشته باشد.
 - اکثر سیاست ها رفتار آینده را بر اساس رفتار گذشته پیش بینی میکنند.
 - سیاست جایگزینی پیچیده تر سربار سخت افزاری و نرم افزاری پیچیده تری برای پیاده سازی لازم دارد.

سیاست جابجایی:

■ قفل کردن قاب:

- صفحه ای که در یک قاب قفل شده باشد نمیتواند جایگزین شود.
- هسته و ساختارهای کنترلی اصلی سیستم عامل در قابهای قفل شده اند.
- میانگیرهای ورودی/خروجی و دیگر نواحی بحرانی میتوانند در قابهای قفل شده قرار گیرند.
- یک بیت قفل به هر قاب پیوند میخورد.

الگوریتم های اصلی جابجایی:

- سیاست بهینه:
- صفحه ای را برای جایگزینی انتخاب میکند که زمان لازم تا مراجعه بعدی به آن طولانی ترین باشد.
- منجر به کمترین تعداد خطای صفحه میشود.
- امکان اجرای این الگوریتم وجود ندارد زیرا نیازمند دانش کافی سیستم عامل از وقایع آینده است.

الگوریتم های اصلی جابجایی:

■ حداقل استفاده در گذشته نزدیک (LRU)

- صفحه ای را جایگزین میکند که برای مدت طولانی دسترسی به آن نشده است.
- بر اساس اصل محلی بودن این صفحه باید صفحه ای باشد که کمترین احتمال مراجعه در آینده نزدیک را داشته باشد.
- هر صفحه را می توان با زمان آخرین مراجعه به آن برچسب گذاری کرد. سربار زیادی برای این روش خواهد بود.

الگوریتم های اصلی جابجایی:

■ خروج به ترتیب ورود:

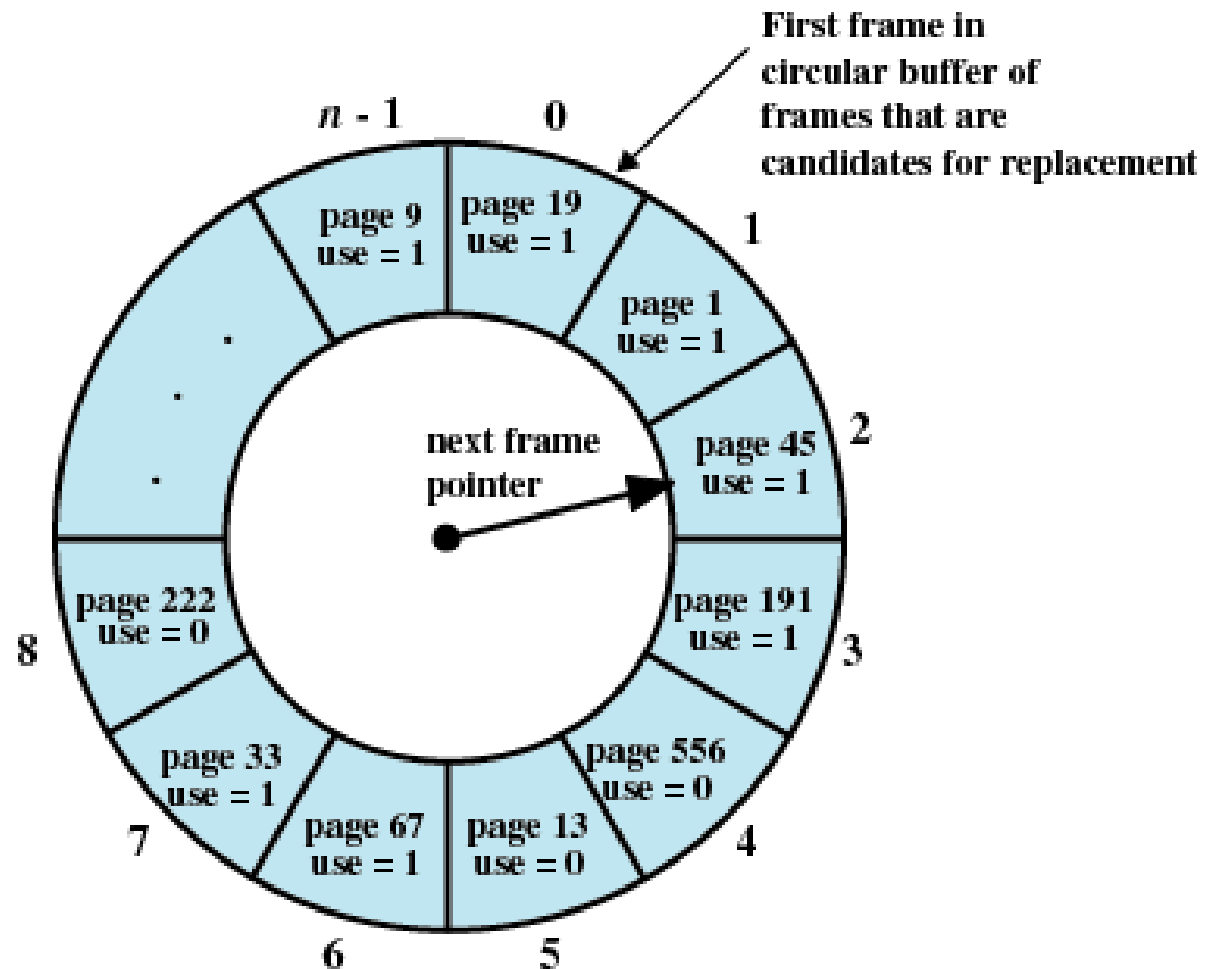
- با قابهای تخصیص یافته به فرایند ها مانند یک میانگیر مدور برخورد میکند.
- صفحات به سبک نوبتی گردش خارج میشوند.
- تنها به یک اشاره گر نیاز دارد، که بطور چرخشی به قابهای فرایند اشاره کند، بنابراین ساده ترین پیاده سازی را دارد.
- صفحه ای که بیشترین مدت در حافظه بوده جایگزین میشود.
- این صفحات ممکن است مجدداً و بزودی نیاز باشند.

الگوریتم های اصلی جابجایی:

■ سیاست ساعت:

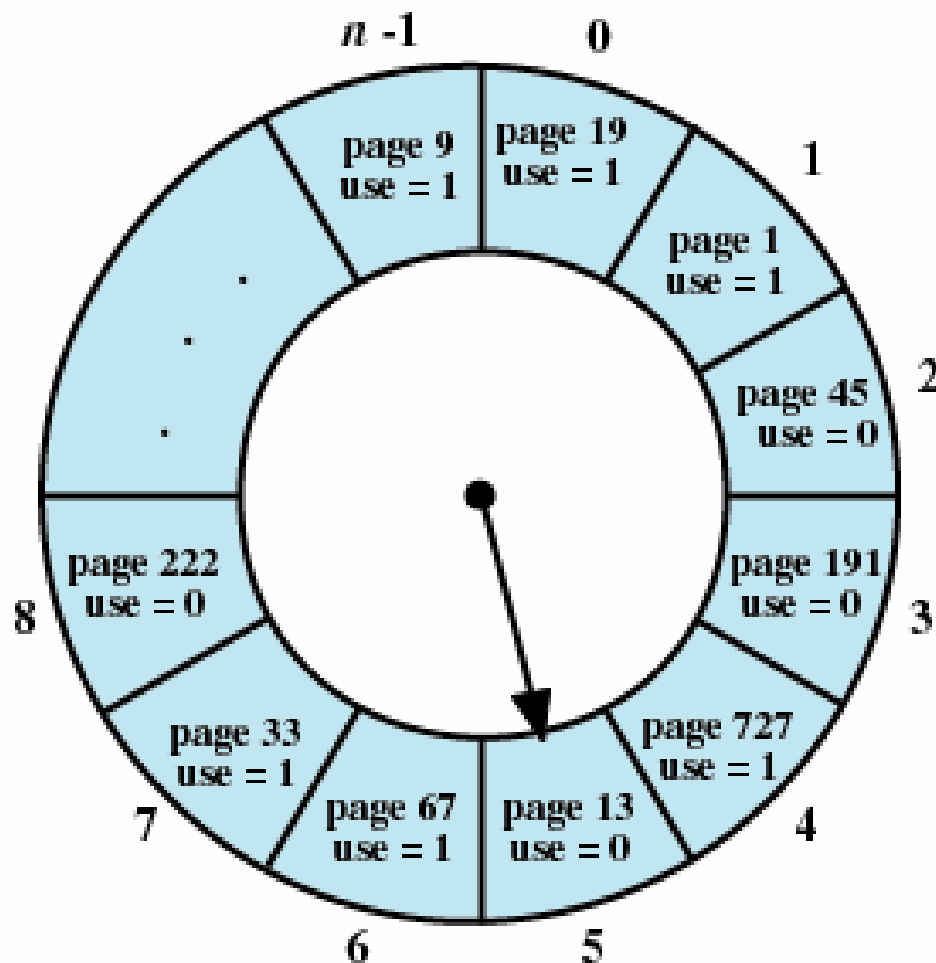
- مانند FIFO است، به جز اینکه در سیاست ساعت از یک بیت اضافی استفاده می شود که به آن بیت استفاده میگویند.
- برای اولین بار که یک صفحه به داخل قابی در حافظه اصلی بار میشود این بیت ۰ میشود
- زمانی که به این صفحه مراجعه میشود این بیت ۱ میشود.
- هنگامیکه زمان جایگزینی صفحه فرا میرسد سیستم عامل میانگیر را مرور کرده تا قابی را بیابد که بیت استفاده از آن صفر باشد.
- در حین جستجو برای جابجایی هر بیت استفاده ۱ به ۰ تغییر میکند.

نمایی از سیاست ساعت (قبل جایگزینی صفحه):



(a) State of buffer just prior to a page replacement

نمایی از سیاست ساعت (بعد جایگزینی صفحه):



(b) State of buffer just after the next page replacement

الگوریتم ساعت با بیت تغییر یافته:

- در کلیه پردازنده هایی که از حافظه مجازی استفاده می کنند یک بیت تغییر برای هر صفحه وجود دارد.
- اگر بیت تغییر و بیت استفاده را با هم در نظر بگیریم هر قاب یکی از دسته های زیر را دارد:
 - اخیرا دستیابی نشده، تغییر نیافته
 - اخیرا دستیابی شده، تغییر نیافته
 - اخیرا دستیابی نشده، تغییر یافته
 - اخیرا دستیابی شده، تغییر یافته
- با این دو بیت الگوریتم به صورت زیر است:
 1. با شروع از موقعیت فعلی اشاره گر، قابها مرور میشود. در حین مرور بیتی تغییر نمیکند. اولین قابی که هر دو بیت استفاده و تغییر آن صفر باشد انتخاب می شود.
 2. در صورت شکست مرحله قبل برای یافتن قابی با بیت تغییر ۱ و بیت استفاده صفر دوباره مرور میکنیم. اولین قاب با این مشخصات انتخاب میشود. در طی این مرحله بیت استفاده هر قابی که مرور می شود صفر می شود.
 3. در صورت شکست مرحله دوم اشاره گر به موقعیت اولیه برگشته است و تمام قابها دارای بیت استفاده صفر هستند. مرحله اول را تکرار می کنیم.

مقایسه الگوریتمهای جایگزینی صفحه در حالت تخصیص ثابت و دیدگاه محلی:

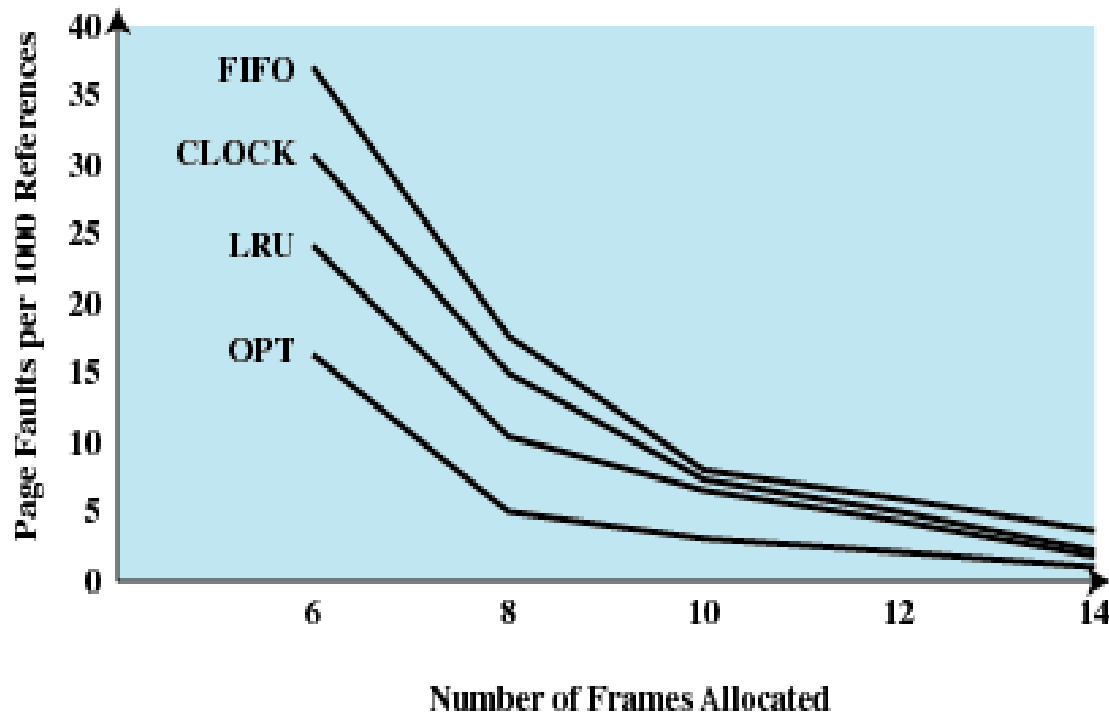


Figure 8.17 Comparison of Fixed-Allocation, Local Page Replacement Algorithms

رویکرد VAX/VMS:

- تعداد کمی از قابها را در تمام مدت آزاد نگه میدارد.
- صفحه جایگزین شده از دست نمی‌رود بلکه به یکی از دو لیست صفحه‌های آزاد و یا صفحه‌های تغییر یافته منتقل می‌شود.
- صفحه به طور فیزیکی جابجا نمی‌شود بلکه مدخل جدول صفحه برای این صفحه حذف می‌شود و در لیست صفحه آزاد یا تغییر یافته قرار می‌گیرد.
- لیست صفحه‌های آزاد، لستی از قابهای موجود برای خواندن صفحه‌ها به داخل است.
- وقتی قرار است صفحه‌ای به داخل خوانده شود، قاب صفحه ابتدای لیست آزاد برای آن اسفاده می‌شود و صفحه‌ای که قبلا در آنجا بود نابود می‌گردد.

مزایای رویکرد VAX/VMS:

- اگر صفحه‌ای که به تازگی به عنوان صفحه خارج شده انتخاب شده است، دوباره فراخوانی گردد با بهای بسیار اندکی به مجموعه مقیم آن فرایند برگردانده می‌شود.
- به جای آن که صفحه‌ها تغییر یافته یکی یکی در حافظه ثانویه نوشته شوند، به صورت گروهی نوشته می‌شوند.

اندازه مجموعه مقیم:

■ عوامل تاثیر گذار در اندازه مجموعه مقیم:

- هر چه حافظه تخصیص یافته به فرآیند کمتر باشد، تعداد فرآیندهای بیشتری می تواند در حافظه قرار بگیرد و کارایی پردازنده بیشتر خواهد شد.
- اگر تعداد نسبتاً کمی از صفحه های یک فرآیند در حافظه اصلی باشد. با وجود اصل محلی بودن نرخ خطای صفحه زیاد می شود.
- به دلیل اصل محلی بودن اختصاص حافظه اصلی بیشتر از حد مشخص تاثیر قابل توجهی بر روی نرخ خطا ندارد.

■ سیاست های اندازه مجموعه مقیم:

- تخصیص ثابت
- تخصیص متغیر: نیاز مند سربار زیاد نرم افزاری است

سیاست پاکسازی

- برعکس سیاست واکنشی است.
 - باید تعیین گردد صفحه تغیر یافته چه موقع در حافظه ثانویه نوشته شود.
 - دوروش معمول :
 - پاکسازی درخواستی :
 - فقط زمانی یک صفحه در حافظه ثانویه نوشته شود که برای جایگزینی انتخاب می شود.
 - پیش پاکسازی
- صفحه های تغیر یافته را قبل از اینکه قابهایشان مورد نیاز باشد در حافظه ثانویه می نویسد و به این ترتیب صفحه ها را می توان به طور دسته ای نوشت.

کنترل بار و سطح چند برنامه‌گی

- کنترل بار در مورد تعیین تعداد فرآیندهایی است که در حافظه اصلی هستند و به آن سطح چند برنامه‌گی گویند.
- با افزایش اندک سطح چند برنامه‌گی انتظار می‌رود که استفاده از پردازنده بیشتر شود. اما به نقطه‌ای خواهیم رسید که مجموعه مقیم برای فرآیندها کافی نخواهد بود و تعداد خطای صفحه زیاد می‌شود و باعث کوبیدگی خواهد شد.

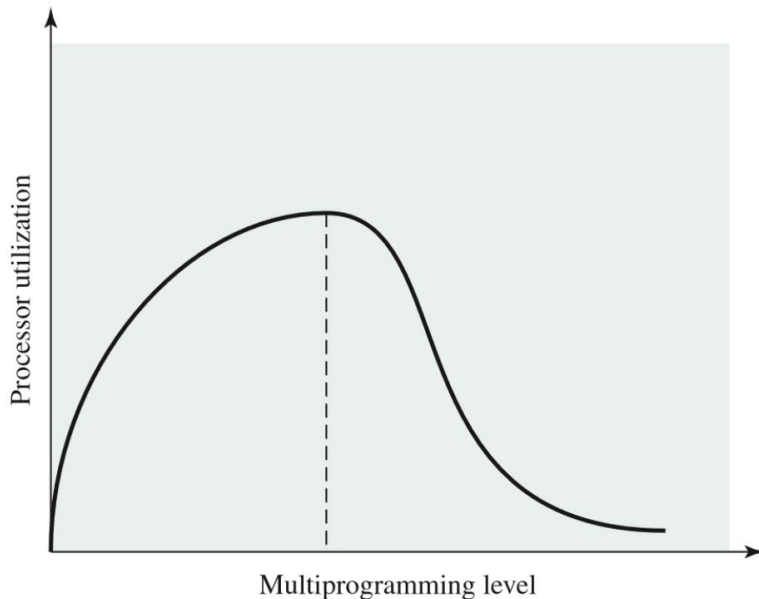


Figure 8.19 Multiprogramming Effects

چند برنامه‌گی و تعلیق فرآیند

- اگر قرار به کاهش سطح چند برنامه‌گی باشد، یکی یا چند فرآیند باید تعلیق شوند.
- سیاست‌ها:
 - فرآیند با کمترین اولویت
 - فرآیند خطا دهنده
 - آخرین فرآیند فعال شده
 - فرآیندی با کوچکترین مجموعه مقیم
 - بزرگترین فرآیند
 - فرآیندی با بزرگترین پنجره اجرایی باقی مانده