

سیستم عامل

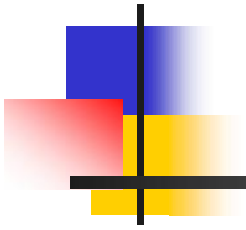
Operating Systems

جلسه دوازدهم

مدرس: اسماعیل طغراعی

www.Teach.Toghraee.ir

حافظه مجازی





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.



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، وقفه ای به معنای خطای دسترسی به حافظه تولید میکند.
- سیستم عامل فرایند وقفه خورده را مسدود میکند و کنترل را بدست میگیرد.

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.
- فرایند میتواند از حافظه بزرگتر باشد.

انواع حافظه:

■ حافظه حقیقی:

■ حافظه اصلی

■ حافظه مجازی:

■ حافظه روی دیسک

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

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 میکند، 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:

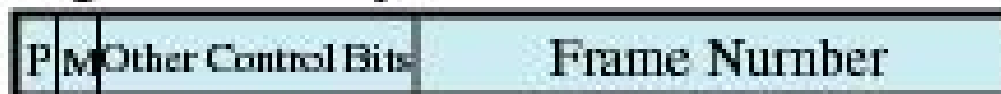
- هر فرایند جدول صفحه مخصوص به خود دارد.
- هر ورودی مدخل صفحه شامل شماره قاب متناظر با صفحه داخل حافظه اصلی است.
- بیت حضور وجود صفحه در حافظه اصلی را نشان میدهد.

صفحه بندی:

Virtual Address



Page Table Entry



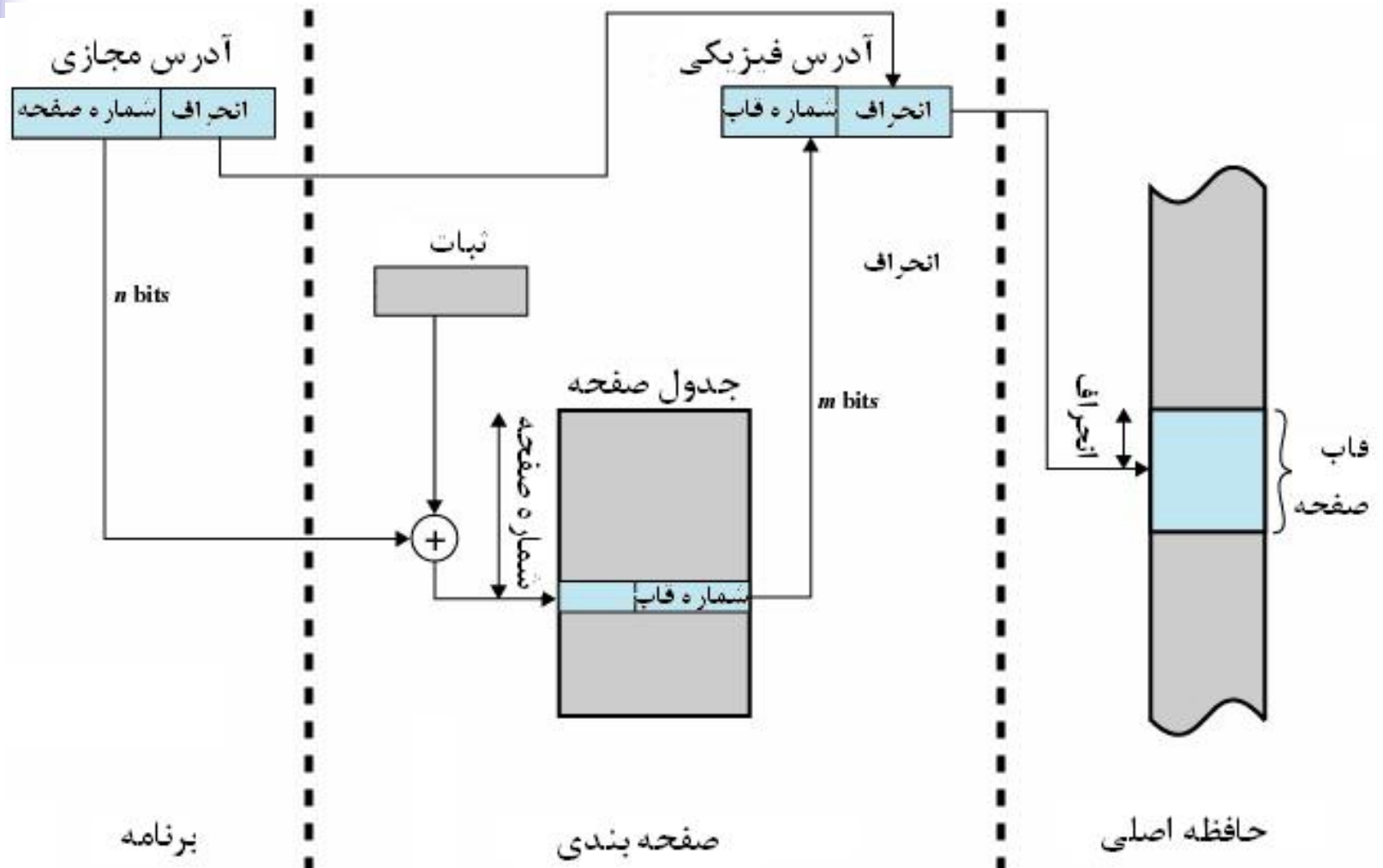
(a) Paging only



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:

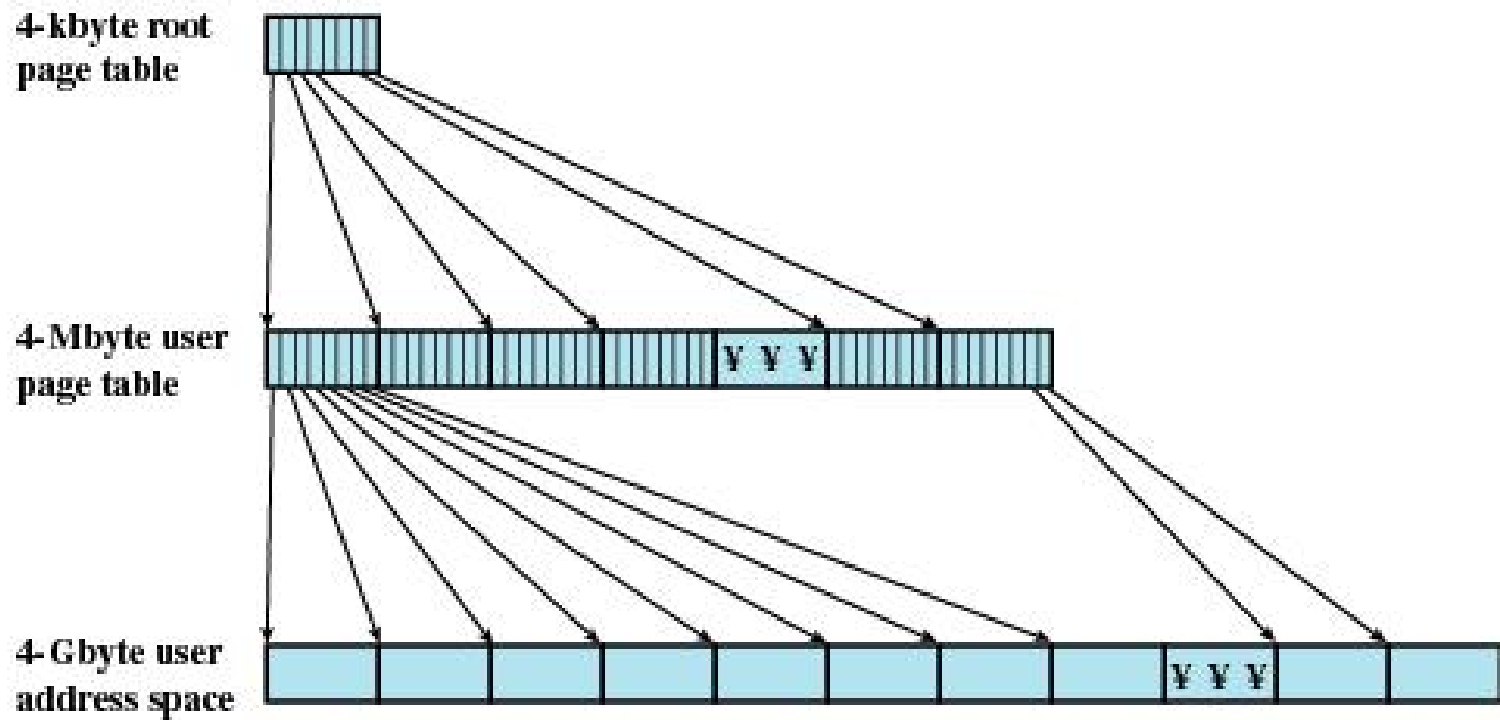


Figure 8.4 A Two-Level Hierarchical Page Table

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.

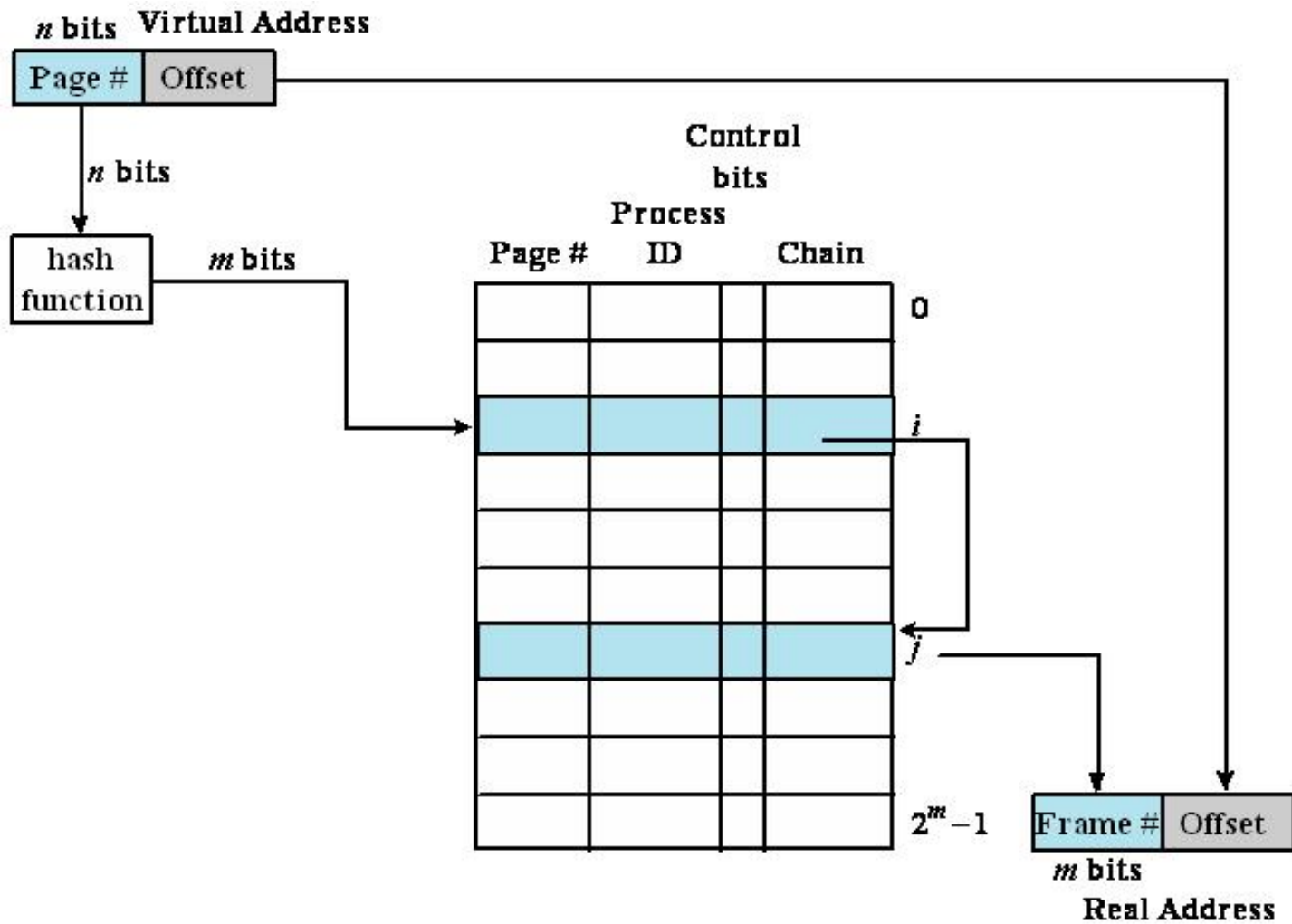


جدول صفحه معکوس:

■ 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.



میانگیر دم دستی ترجمه:

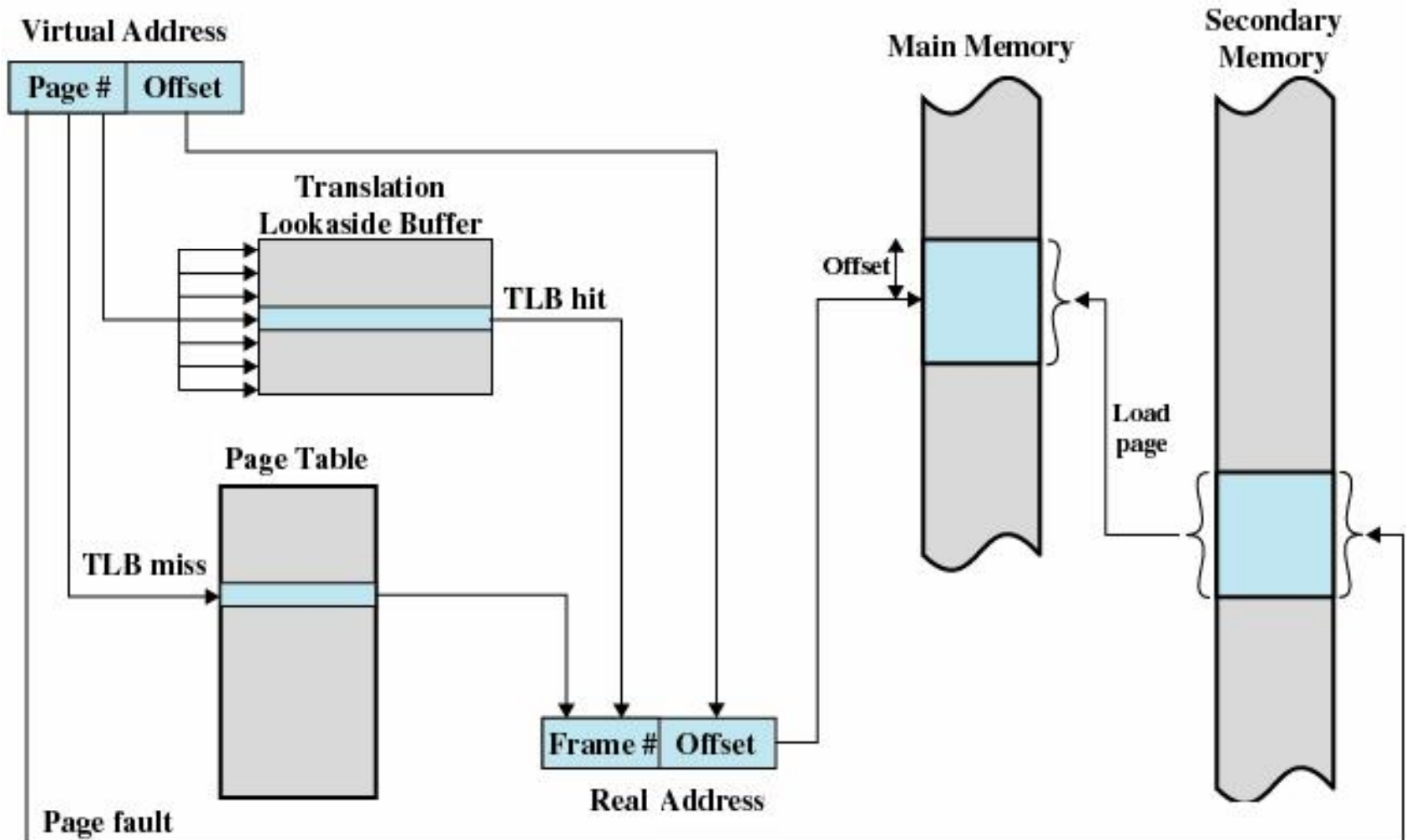
- حاوی مدخلهایی از جدول صفحه است که اخیراً به آنها دسترسی شده است.
- با داشتن یک آدرس مجازی پردازنده میانگیر دم دستی را امتحان میکند.
- اگر مدخل مورد نظر جدول صفحه پیدا شد(اصابت)، شماره قاب بازیابی شده و آدرس حقیقی شکل میگیرد.
- اگر مدخل مورد نظر از جدول صفحه پیدا نشد(عدم اصابت) پردازنده از شماره صفحه به عنوان شاخص جدول صفحه فرایند استفاده میکند. و مدخل مورد نظر در جدول صفحه را آزمایش میکند.



میانگیر دم دستی ترجمه:

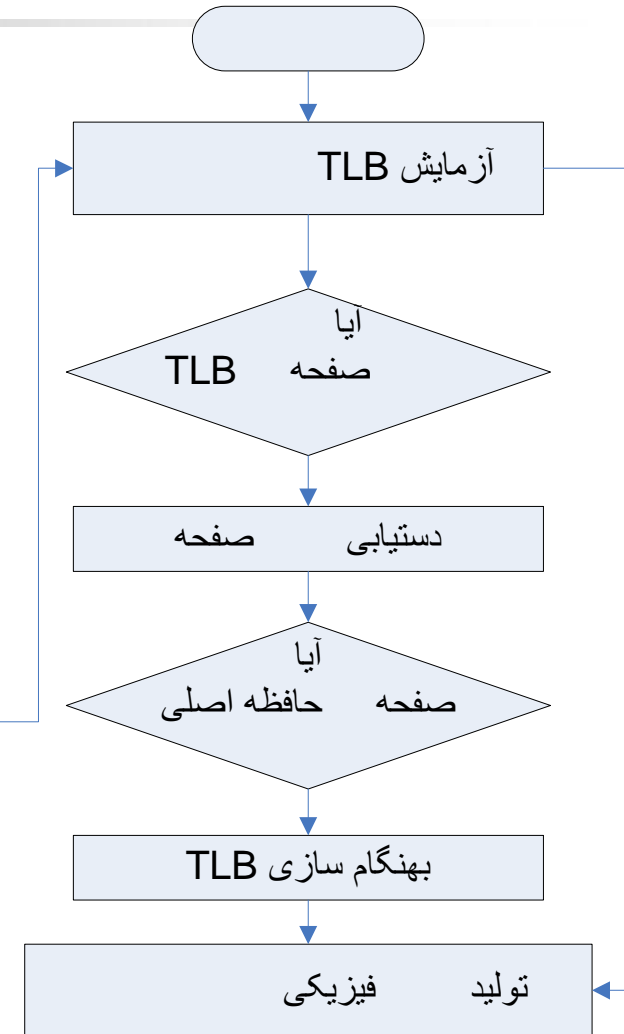
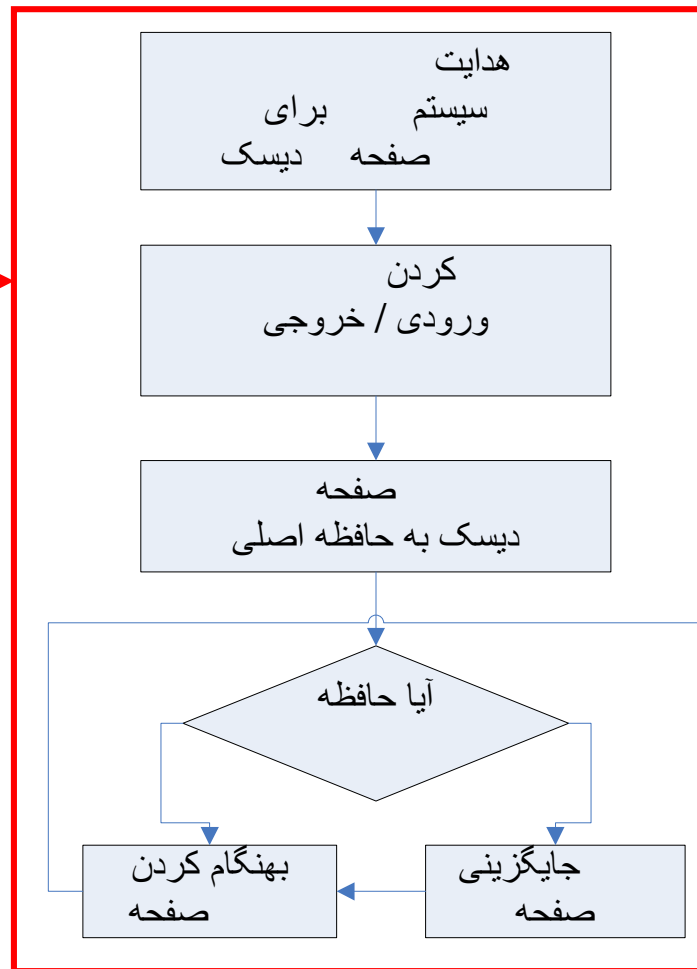
- بیت حضور (حضور صفحه در حافظه) بررسی میشود:
- اگر در حافظه اصلی نبود خطای فقدان صفحه (Page Fault) صادر میشود.
- اگر در حافظه اصلی بود، پردازنده شماره قاب را از مدخل جدول صفحه برای تشکیل آدرس حقیقی بدست می آورد و میانگیر دم دستی را بهنگام میکند تا شامل این مدخل جدید صفحه شود.
- به طور کلی هدف میانگیر دم دستی ترجمه کم کردن تعداد مراجعه به دیسک برای واکنشی مدخل جدول صفحه است.

استفاده از میانگیر دم دستی ترجمه:



استفاده از میانگیر دم دستی ترجمه:

میانگیر
دم دستی
ترجمه





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 جابجایی:

- 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 جایابی:

■ 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.

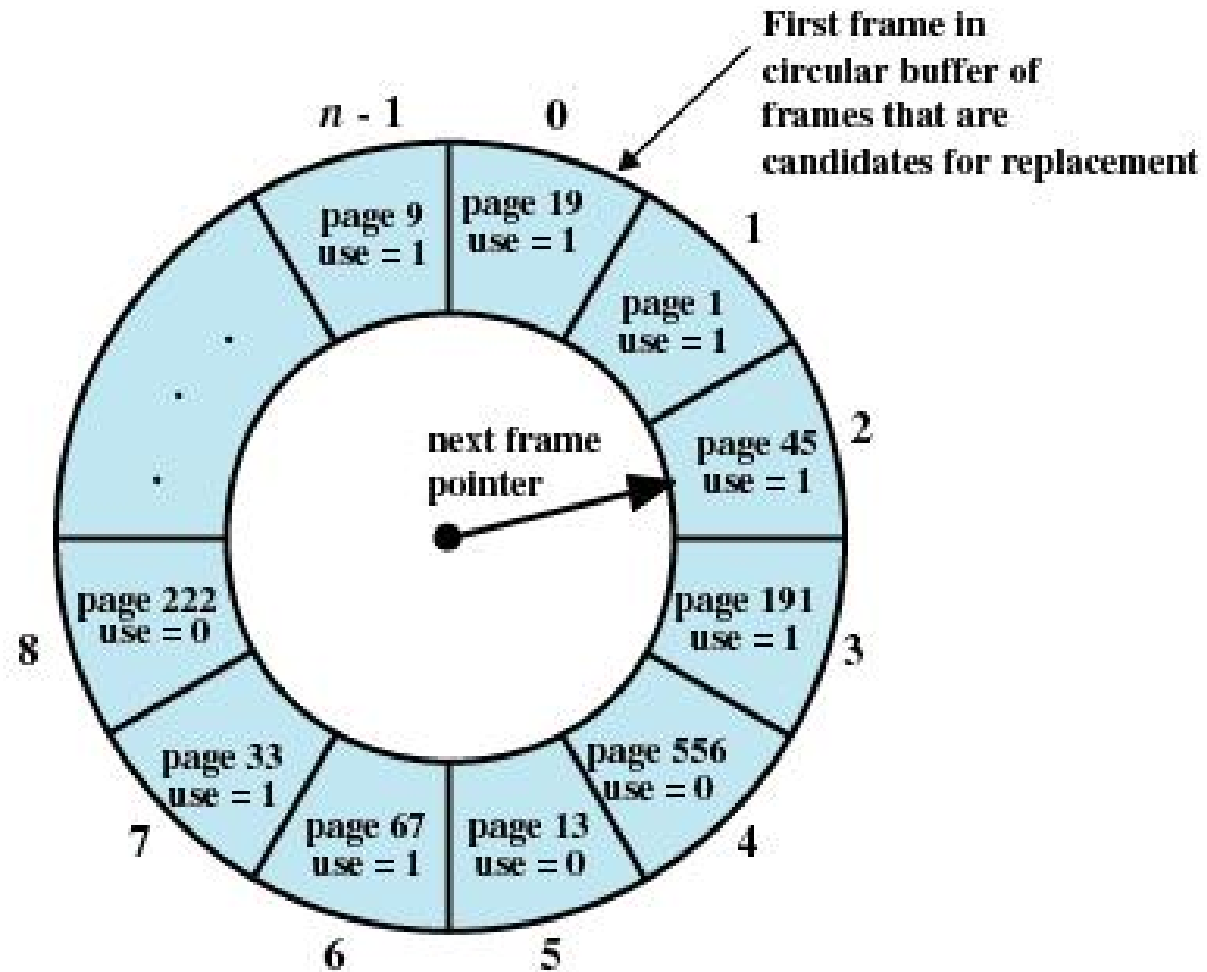


الگوریتم های اصلی جابجایی:

■ سیاست ساعت:

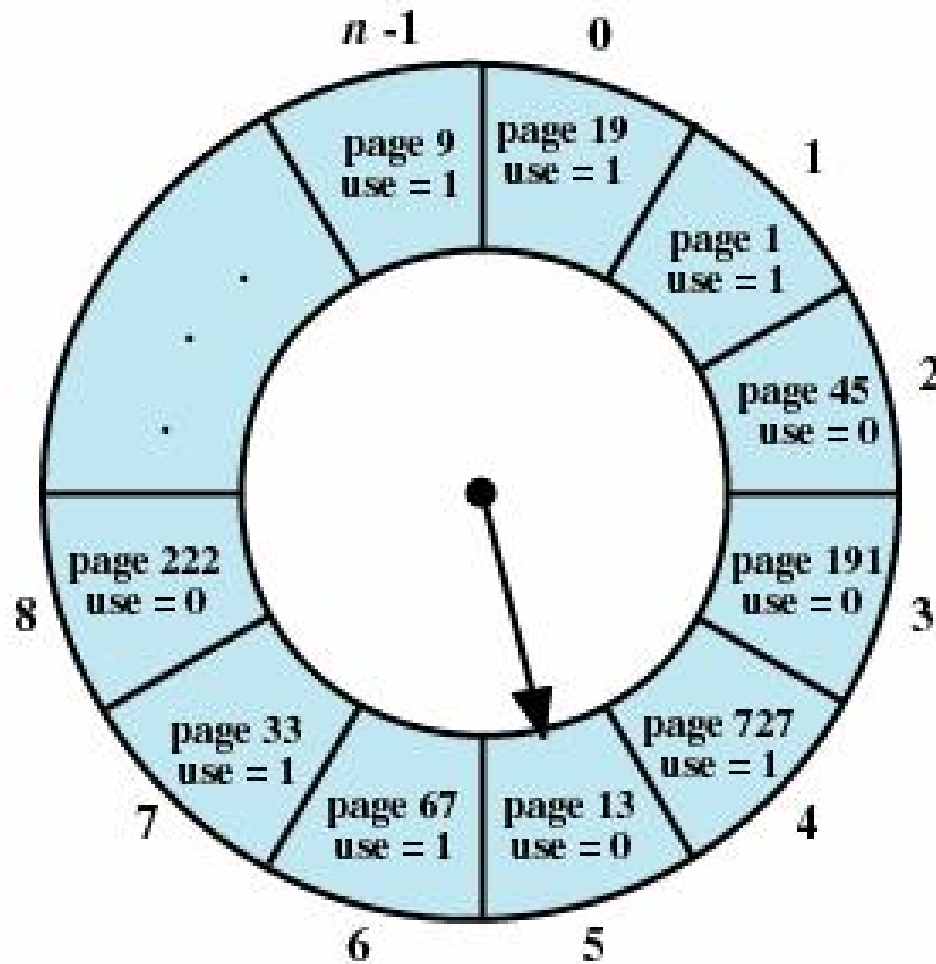
- مانند FIFO است، به جز اینکه در سیاست ساعت از یک بیت اضافی استفاده می شود که به آن بیت استفاده میگویند.
- برای اولین بار که یک صفحه به داخل قابی در حافظه اصلی بار میشود این بیت ۰ میشود
- زمانی که به این صفحه مراجعه میشود این بیت ۱ میشود.
- هنگامیکه زمان جایگزینی صفحه فرا میرسد سیستم عامل میانگیر را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

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

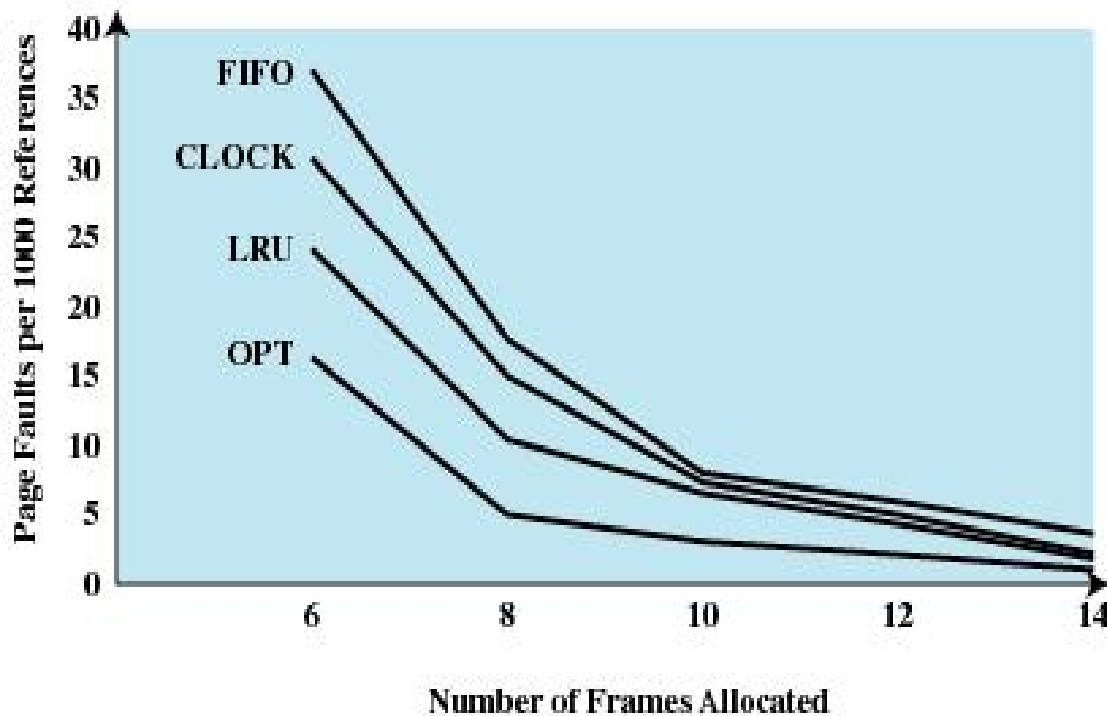


Figure 8.17 Comparison of Fixed-Allocation, Local Page Replacement Algorithms