دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکدهٔ مهندسی کامپیوتر و فنآوری اطلاعات

اصول و مفاهیم سیستم عامل

ترم بهمن ۹۷

فصل پنجم: حافظهٔ مجازی

نستوه طاهری جوان

nastooh@aut.ac.ir

حافظهٔ مجازی

- ایدهٔ اصلی حافظهٔ مجازی:
  - حتی اگر به اندازهٔ کافی فضای خالی بر روی حافظه نداشته باشیم، باز هم به یک فرآیند اجازهٔ اجرا بدهیم!
    - نکتهٔ اصلی: یک فرآیند در آنِ واحد به همهٔ داده ها و کد خود نیاز ندارد، بلکه در هر لحظه فقط به بخشی از داده و قسمتی از کد نیاز دارد.
    - در این تکنیک فقط قسمتی از فرآیند به حافظه آورده می شود که در حال حاظر به آن نیاز است و مابقی فرآیند کماکان بر روی دیسک قرار می گیرد.
    - در طول اجرای فرآیند ناگزیر از جابجایی های مکرر بین دیسک و حافظه هستیم.
    - با این کار می توانیم فرآیندهای بیشتری را به حافظه دعوت کنیم.
    - این تکنیک معمولا از دید برنامه نویس (و فرآیند) پنهان است. در این حالت فرآیندها گمان می کنند به طور کامل در حافظه قرار دارند، غافل از اینکه بخش اعظم هر یک از آن ها بر روی دیسک قرار دارد.
    - در این حالت **مجموع** کلیهٔ فضای آدرس فرآیندهای در حال اجرا، می تواند از اندازهٔ حافظهٔ فیزیکی بزرگتر شود، و این یعنی **حافظهٔ مجازی!**

nastooh@aut.ac.ir

صفحه بندی برحسب نیاز

- ایدهٔ حافظهٔ مجازی معمولا با تکنیک صفحه بندی پیاده سازی می شود که به آن Demand Paging گویند.
  - پیاده سازی حافظهٔ مجازی با تکنیک هایی مانند قطعه بندی، پیچیدگی زیادی دارد.
  - در این روش باید مشخص شود کدام صفحات بر روی دیسک و کدام صفحات بر روی حافظه قرار دارند، برای این منظور عموماً از یک بیت اعتبار در جداول صفحه استفاده می کنند.
  - اگر یک فرآیند به صفحه ای نیاز داشت که در حافظه وجود ندارد (بیت اعتبار آن برابر صفر بود) یک وقفهٔ Page Fault رخ می دهد. در این حالت سیستم عامل باید یک قاب برای این صفحه تهیه کرده و آن را به حافظه منتقل کند.

nastooh@aut.ac.ir

صفحه بندی برحسب نیاز

- آدرسی که توسط فرآیند مورد ارجاع قرار می گیرد، آدرس مجازی و آدرس های حافظه، آدرس حقیقی نام دارند.
- هنگام بروز وقفهٔ نقص صفحه، سیستم عامل باید هر چه سریع تر آن صفحه را به یکی از قاب های حافظه منتقل کند. در این حالت اگر قاب آزاد وجود نداشته باشد، یکی از صفحات باید به دیسک منتقل شود.
  - نکتهٔ مهم چگونگی انتخاب یک صفحه برای ترک حافظه است. این انتخاب تاثیر مستقیمی بر روی کارآیی و تعداد وقفه های نقص صفحه در آینده دارد.
- یکی از نکات مهم تخصیص قاب به فرآیندهاست
  - تخصیص مساوی در برابر تخصیص متناسب
    - آیای به همهٔ فرآیندها به تعداد برابر قاب تخصیص دهیم؟
  - تخصیص ثابت در برابر تخصیص متغییر
    - آیا می توان در حین اجرا تعداد قاب های یک فرآیند را تغییر داد؟
  - جایگزینی محلی در برابر جایگزینی سراسری
    - هنگام نقص صفحه می توان از قاب های فرآیندهای دیگر استفاده کرد؟

nastooh@aut.ac.ir

صفحه بندی برحسب نیاز

- انتخاب اندازهٔ قاب (صفحه) ها بر عهدهٔ سیستم عامل است و تاثیر مستقیمی بر کارآیی دارد.
  - هرچه اندازهٔ قاب کوچکتر باشد، تکه تکه شدن داخلی به ازای آخرین قاب یک فرآیند کم تر می شود.
  - بزرگ بودن اندازهٔ صفحه یعنی بلااستفاده ماندن بخش بزرگی از حافظه. زیرا وقتی یک صفحهٔ بزرگ به حافظه دعوت می شود، تمام قسمت های آن مورد استفاده نیستند.
  - کوچک شدن اندازهٔ صفحات باعث افزایش اندازهٔ جداول صفحه می شود.
  - کوچک شدن اندازهٔ صفحات به معنای افزایش تعداد دفعات جابجایی بین دیسک و حافظه است.
- سربار حافظه به ازای هر فرآیند تا اینجا به دو بخش تقسیم می شود، یکی میزان حافظهٔ تلف شده به ازای آخرین صفحه فرآیند و دیگری اندازهٔ جدول صفحهٔ فرآیند.
  - اگر اندازهٔ صفحات را کوچک بگیریم، قسمت اول کاهش و قسمت دوم افزایش می یابد، و بالعکس. آیا می توان یک مقدار بهینه برای آن تعین کرد؟؟؟

nastooh@aut.ac.ir

صفحه بندی برحسب نیاز

- اگر تعداد قاب های تخصیص داده شده به یک فرآیند آنقدر کم باشد که آن فرآیند نتواند صفحاتی که **زیاد** با آنها سر و کار دارد را به حافظه منتقل کند، سرعت اجرای آن فرآیند به شدت کاهش می یابد و تعداد خطاهای نقص صفحهٔ آن چندین برابر حالت عادی می شود. به این حالت کوبیدگی یا Thrasging گویند.
  - نکتهٔ مهم این است که هنگام کوبیدگی، بهره وری پردازنده به شدت کاهش می یابد، زیرا زمان سیستم صرف انتقال های پی در پی صفحات بین حافظه و دیسک می شود.
  - مهار کردن پدیدهٔ کوبیدگی دقت فراوانی نیاز دارد. به عنوان مثال از آنجایی که هنگام کوبیدگی، بهرهٔ مفید پردازنده کاهش می یابد (پردازنده قالبا منتظر است)، زمانبند کار ممکن است برای افزایش بهرهٔ مفید پردازنده، درجهٔ چند برنامگی را افزایش دهد (یعنی چند کار دیگر را برای اجرا به حافظه دعوت کند) که این امر ممکن است منجر به تشدید کوبیدگی و حتی سرایت آن به برنامه های دیگر شود.

nastooh@aut.ac.ir

صفحه بندی برحسب نیاز

- رعایت اصول سادهٔ برنامه نویسی می تواند در افزایش سرعت اجرای برنامه و کاهش تعداد نقص صفحه تاثیر مستقیم گذارد.
  - فرض کنید یک آرایهٔ دوبعدیِ ۱۰۰ در ۱۰۰ عنصری داریم که هر عنصر ۲ بایت و اندازهٔ هر صفحه ۲۰۰ بایت است. در واقع این آرایهٔ ۱۰ هزار عنصری در ۱۰۰ صفحه قرار دارد. می دانیم در زبانهایی مانند سی و پاسکال، آرایه ها به صورت سطری در حافظه قرار می گیرند. فرض کنیم **فقط یک قاب برای داده ها** در اختیار این برنامه باشد، حال دو قطعه کد زیر را از نظر تعداد خطای نقص صفحه با هم مقایسه کنید:

```
for i:=1 to 100 do
    for j:=1 to 100 do
        a[i,j]:=0;
```

```
for i:=1 to 100 do
    for j:=1 to 100 do
        a[j,i]:=0;
```


nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

- ✓ الگوریتم های جایگزینی صفحه
  - ○ الگوریتم FIFO

- در این روش سیستم عامل لیستی از صفحات را به ترتیب ورود به حافظه نگهداری می کند.
- وقتی یک خطای نقص صفحه رخ می دهد، سیستم عامل **قدیمی ترین** صفحه را برای بیرون رفتن انتخاب می کند.

nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

- مثال از FIFO

- فرض کنید در سیستمی ۳ قاب حافظه وجود دارد، اگر درخواست ها از چپ به راست به این سیستم وارد شوند، چند وقفهٔ نقص صفحه رخ می دهد؟

4,3,2,1,4,3,5,4,3,2,1,5

ورودی	۴	۳	۲	۱	۴	۳	۵	۴	۳	۲	۱	۵
قاب ۱	۴	۴	۴	۱	۱	۱	۵	۵	۵	۵	۵	۵
قاب ۲		۳	۳	۳	۴	۴	۴	۴	۴	۲	۲	۲
قاب ۳			۲	۲	۲	۳	۳	۳	۳	۳	۱	۱
نقص صفحه	x	x	x	x	x	x	x			x	x	

nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

- در نگاه اول به نظر می رسد با افزایش قاب های در اختیار یک فرآیند، تعداد نقص صفحه ها همواره کاهش می یابد، اما در الگوریتم FIFO (و برخی از ایده های دیگر) با افزایش تعداد قاب ها، تعداد قفه های نقص صفحه افزایش می یابد. به این پدیده، ناهنجاری بی لِیدی (Belady Anomaly) گویند.
- مثال: همان حالت قبل با چهار قاب حافظه

ورودی	۴	۳	۲	۱	۴	۳	۵	۴	۳	۲	۱	۵
قاب ۱	۴	۴	۴	۴	۴	۴	۵	۵	۵	۵	۱	۱
قاب ۲		۳	۳	۳	۳	۳	۳	۴	۴	۴	۴	۵
قاب ۳			۲	۲	۲	۲	۲	۲	۳	۳	۳	۳
قاب ۴				۱	۱	۱	۱	۱	۱	۲	۲	۲
نقص صفحه	x	x	x	x			x	x	x	x	x	x

nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

- الگوریتم Optimal

- در این ایده صفحه ای باید حافظه را ترک کند که در آینده، دیرتر از همه به آن نیاز داریم.
- در این ایده خطاهای نقص صفحه به حداقل می رسد.
- این ایده ناهنجاری بی لِیدی را ندارد.
- ای این ایده فقط می توان برای ارزیابی دیگر الگوریتم ها استفاده کرد، زیرا در عمل قابل پیاده سازی نیست. (سیستم عامل نمی داند در آینده فرآیندها به چه ترتیبی به صفحات خود نیاز دارند.)

nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

- مثال از Optimal
  - حافظه ای را با سه قاب در نظر بگیرید، با استفاده از روش بهینه و برای دنبالهٔ ارجاعات زیر، چند خطای نقص صفحه رخ می دهد؟

3,2,1,4,3,1,2,5,3,4,1

ورودی	۳	۲	۱	۴	۳	۱	۲	۵	۳	۴	۱
قاب ۱	۳	۳	۳	۳	۳	۳	۳	۳	۳	۳	۱
قاب ۲		۲	۲	۴	۴	۴	۴	۴	۴	۴	۴
قاب ۳			۱	۱	۱	۱	۲	۵	۵	۵	۵
نقص صفحه	x	x	x	x			x	x			x

nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

- الگوریتم LRU(Least Recently Used)

- ایدهٔ اصلی این روش این است که اگر صفحه ای اخیراً مراجعات زیادی داشته، به احتمال قوی در دستورات بعدی هم مراجعات زیادی خواهد داشت، بالعکس اگر یک صفحه اخیراً هیچ ارجاعی نداشته، احتمالا در آینده نیز ارجاعی نخواهد داشت!
- در واقع این الگوریتم همانند الگوریتم بهینه است، منتها به جای توجه به آینده، به گذشته چشم دوخته است.
- این الگوریتم ناهنجاری Belady ندارد.
- برای پیاده سازی عملی این ایده باید لیستی از صفحات در اختیار داشته باشیم که در آن لیست، صفحات به ترتیب ارجاعات اخیر مرتب شده باشند. (هزینهٔ این پیاده سازی قدری بالاست. زیرا این لیست با هر بار ارجاع باید به روز شود. برای این منظور می توان از لیست های پیوندی دو طرفه به شیوهٔ موثری استفاده کرد.)

nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

- مثال از LRU

- حافظه ای را با ۳ قاب در نظر بگیرید. با استفاده از ایدهٔ LRU و برای دنبالهٔ ارجاعات زیر، چند خطای نقص صفحه رخ می دهد؟

3,1,2,4,3,4,2,4,1,3,5,1,3,2

ورودی	۳	۱	۲	۴	۳	۴	۲	۴	۱	۳	۵	۱	۳	۲
قاب ۱	۳	۳	۳	۴	۴	۴	۴	۴	۴	۴	۵	۵	۵	۲
قاب ۲		۱	۱	۱	۳	۳	۳	۳	۱	۱	۱	۱	۱	۱
قاب ۳			۲	۲	۲	۲	۲	۲	۲	۳	۳	۳	۳	۳
نقص صفحه	x	x	x	x	x				x	x	x			x

nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

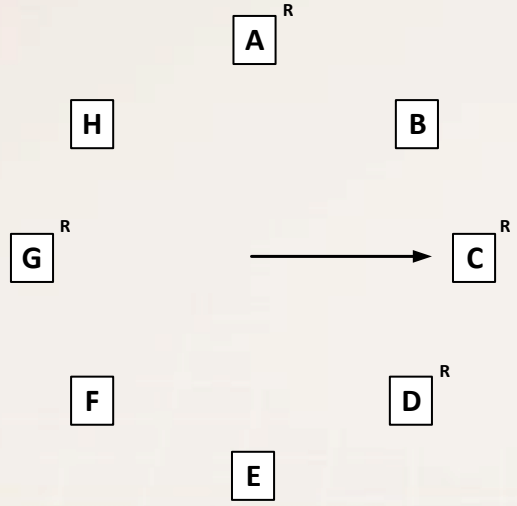
- الگوریتم Second Chance

- این ایده از خانوادهٔ FIFO است، با این تفاوت که صفحاتی که زیاد استفاده شده اند، از حافظه خارج نشوند.
- در این ایده به ازای هر صفحه در جدول صفحه یک بیت با عنوان R اضافه می شود. اگر به صفحهٔ موجود در حافظه ارجاع شود، آن بیت یک می شود.
- در این ایده برای حذف، به دنبال قدیمی ترین صفحه می رویم، اما قبل از حذف، بیت R آن را چک می کنیم، اگر این بیت صفر باشد، به این معنی است که این صفحه هم قدیمی است و هم ارجاعی به آن صورت نگرفته است، پس حذف می شود. اما اگر بیت R آن یک بود، بیت آن صفر شده و به انتهای صف فرستاده می شود. در واقع با این ایده یک شانس دوباره به آن داده می شود.
- این ایده ناهنجاری Belady را دارد.
- در این ایده اگر صفحه ای زیاد مورد ارجاع قرار بگیرد، این شانس را دارد که در حافظه باقی بماند. زیرا پس از صفر شدن بیت Rاش، و تا زمانی که دوباره به ابتدای صف برسد اگر به آن ارجاع شود، بیت Rش دوباره یک می شود!

nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

- الگوریم ساعت (Clock)
  - این الگوریتم در واقع یک ایده برای پیاده سازی روش دومین شانس است.
  - در این روش در واقع از یک لیست پیوندی حلقوی استفاده می کنیم که مانند یک ساعت عقربه ای عمل می کند.
  - در این ساعت عقربه همواره قدیمی ترین صفحه را نشان می دهد.
  - هنگام خطای نقص صفحه، صفحه ای که عقربه روی آن است بررسی می شود، اگر بیت R آن صفر بود، حذف می شود، اما اگر بیت R آن یک بود، ابتدا صفر شده و سپس عقربه یک واحد جلو می رود.



ایدهٔ ساعت به صورت شماتیک

nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

- الگوریتم سالمندی (Aging)

- در این روش به ازای هر صفحه یک شمارنده (مثلا هشت بیتی) تعبیه می شود. همهٔ شمارنده ها در پریودهای زمانی منظم یک بیت به سمت راست شیف داده می شوند و جای خالی بیت سمت چپ با بیت R صفحه ها پر می شود.
- پس از هر شیفت، بیت R صفر می شود.
- می توان گفت این شمارنده ها تاریخچهٔ ارجاعات را نگهداری می کنند.
- سیستم عامل هنگام نقص صفحه، صفحه ای را انتخاب می کند که شمارندهٔ کمتری دارد.
- شمارندهٔ کوچکتر در واقع به این معناست که این صفحه اخیرا کمتر استفاده شده است.
- در این الگوریتم صفحه ای که شمارندهٔ آن 10000001 است، اولویت بیشتری برای ماندن در حافظه نسبت به صفحه ای با شمارندهٔ 01111111 دارد!
- هنگامی که شمارنده ها با هم برابرند، سیستم از ایدهٔ FIFO استفاده می کند.
- این روش در واقع نوعی پیاده سازی برای ایدهٔ LRU است.

nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

- الگوریتم NRU(Not Recently Used)

- در این روش برای هر صفحه از دو بیت وضعیت استفاده می شود. M و R (مخفف Modified و Referenced)
- بیت R هنگامی برای یک صفحه ست می شود که به آن صفحه ارجاع شده باشد. (چه برای خواندن و چه برای نوشتن)
- هنگام تغییر یک صفحه، بیت M آن ست می شود.
- برای ایجاد تمایز بین صفحاتی که اخیرا به آنها ارجاع شده و صفحاتی که قبلا به آنها ارجاع شده، در پریودهای منظم بیت های R صفر می شوند.
- هنگام نقص صفحه سعی می شود صفحه ای خارج شود که به آن ارجاعی صورت نگرفته است.
- اگر صفحه ای بدون ارجاع پیدا نشد، باید از بین صفحاتی که به آنها ارجاع شده است، صفحه ای را انتخاب کنیم که تغییر نکرده باشد (زیرا به نوشتن دوبارهٔ آن بر روی دیسک نیاز نیست)
- این ایده احتمال ناهنجاری Belady را دارد.

nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

- الگوریتم NRU (ادامه)
  - در واقع این ایده صفحات را به چهار گروه زیر تقسیم می کند.

رده	R	M
گروه 0	0	0
گروه 1	1	0
گروه 2	0	1
گروه 3	1	1

  - گروه صفر صفحاتی هستند که به آنها ارجاع نشده.
  - گروه یک صفحاتی هستند که برای خواندن به آنها ارجاع شده.
  - گروه دو صفحاتی هستند که برای نوشتن به آنها ارجاع شده اما با منقضی شدن تایمر، بیت R آنها صفر شده.
  - گروه سه صفحاتی هستند که برای نوشتن به آنها اخیرا ارجاع شده.
  - بهترین گزینه برای ترک حافظه به ترتیب گروه های صفر، بعد یک، بعد دو و در نهایت سه هستند.

nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

- الگوریتم LFU(Least Frequently Used)
  - نام دیگر: NFU(Not Frequently Used)
  - برای هر صفحه یک شمارنده در نظر میگیرند. سیستم عامل به صورت دوره ای بیت R مربوط به هر صفحه را به شمارندهٔ آن اضافه می کند و بیت R را صفر می کند.
  - در واقع شمارنده تعداد ارجاعات این صفحه را شمارش می کند.
  - هر صفحه ای که شمارندهٔ آن کمتر بود، برای خروج انتخاب می شود.
  - این روش درواقع ایده ای برای پیاده سازی LRU می باشد.
  - این الگوریتم در واقع چیزی را فراموش نمیکند،
    - به عنوان مثال ممکن است یک صفحه در یک بازهٔ کوتاه ارجاعات زیادی داشته باشد، پس شمارندهٔ آن به ناگاه افزایش می یابد. اما به ناگاه ارجاعات به این صفحه قطع می شوند. در این ایده این صفحه کماکان در حافظه می ماند.

nastooh@aut.ac.ir

الگوریتم های جایگزینی صفحه

- الگوریتم MFU(Most Frequently Used)
  - این روش همانند LFU یک شمارنده برای هر صفحه استفاده می کند.
  - هنگام نقص صفحه، صفحه ای برای خروج انتخاب می شود که شمارندهٔ بزرگتری داشته باشد!
  - در واقع ایدهٔ این الگوریتم این است که صفحه ای که شمارندهٔ بزرگتری دارد، به اندازهٔ کافی در حافظه بوده است و باید شانس استفاده شدن به دیگران داده شود.

nastooh@aut.ac.ir

تمرین ها

- ✓ تمرین ۱: دربارهٔ مدیریت حافظهٔ مجازی در ویندوز تحقیق کنید.
  - منبع [3]، بخش 5-8.
- ✓ تمرین ۲: الگوریتم جایگزینی صفحه «ساعت دو عقربه ای» که در سیستم عامل یونیکس (SVR4) استفاده می شود را بررسی کنید.
  - منبع [3]، شکل 8-23.
- ✓ تمرین ۳: دربارهٔ روال مدیریت حافظه مبتنی بر معماری پنتیوم از شرکت اینتل تحقیق کنید.
  - مرجع [2]، بخش 3-7-3

nastooh@aut.ac.ir

تمرین ها

- تمرین ۴: فرض کنید سربار حافظه به ازای هر فرآیند به دو بخش تقسیم می شود، یکی میزان حافظهٔ تلف شده به ازای آخرین صفحه فرآیند و دیگری اندازهٔ جدول صفحهٔ فرآیند. اگر اندازهٔ صفحات را کوچک بگیریم، قسمت اول کاهش و قسمت دوم افزایش می یابد، و بالعکس. آیا می توان یک مقدار بهینه برای اندازهٔ صفحه تعیین کرد؟؟؟
  - مرجع [2]، بخش 3-5-3
- تمرین ۵: دربارهٔ پیاده سازیِ حافظهٔ مجازی با تکنیک قطعه بندی تحقیق کنید.
  - مرجع [3]، خلاصه جدول 2-8

nastooh@aut.ac.ir

تمرین ها

- ✓ تمرین ۶: دربارهٔ اشتراک گذاری حافظه بین فرآیندها تحقیق کنید.
  - منبع [1]، بخش 4-5-8 برای صفحه بندی.
  - منبع [3]، صفحه ۳۵۲ برای قطعه بندی.
- ✓ تمرین ۷: دربارهٔ تفاوت های سیستم عامل های ۳۲ بیتی و ۶۴ بیتی در مبحث مدیریت حافظه به طور **دقیق** تحقیق کنید.
  - منبع: اینترنت (آزاد)
- ✓ تمرین ۸: دربارهٔ Prepaging یا پیش صفحه بندی تحقیق کنید. چه زمانی پیش صفحه بندی مقرون به صرفه است؟
  - مرجع [1]، بخش 1-9-9

nastooh@aut.ac.ir

تمرین ها

- تمرین ۹: با توجه به روش های معرفی شده جهت جایگزینی صفحه در حافظه مجازی، روش پیشنهادی (ابداعی) خود را ارائه داده و در مورد مزایا و معایب احتمالی آن بحث کنید. سعی کنید حتما به جنبه ها و نیازمندی های پیاده سازی عملی روش خود نیز بپردازید.

nastooh@aut.ac.ir

نمونه تست های کنکور ارشد

✓ ارشد، آزاد، ۸۵

- سیستمی از تکنیک صفحه بندی مجازی استفاده می کند. اگر حداکثر فضای آدرس مجازی که یک برنامه می تواند به آن ارجاع نماید، برابر با 2n و اندازهٔ هر صفحه برابر با 2p باشد، انگاه تعداد صفحات فضای مجازی کدام است؟
- الف) 2^{n+p} ب) n-p ج) n+p د) 2^{n-p}

✓ ارشد، دولتی ۷۷

- یک سیستم حافظه صفحه بندی مجازی را در نظر بگیرید که حداکثر اندازهٔ جدول صفحه هر فرآیند در آن برابر ۸ گیگا بایت، اندازهٔ صفحه برابر ۸ کیلو بایت و مدخل های هر جدول صفحه ۱۶ بیتی باشند. آدرس های ماشینی که بتواند این حافظهٔ مجازی را مستقیما آدرس دهی کند، باید چند بیتی باشد؟

الف)۳۲ ب) ۱۶ ج) ۸ د) ۲۴

nastooh@aut.ac.ir

نمونه تست های کنکور ارشد

✓ ارشد، آزاد، ۸۵

- سیستمی از الگوریتم جایگزینی LFU استفاده میکند. اگر سیستم دارای ۴ قاب خالی باشد و برنامه ای به ترتیب از چپ به راست به صفحات مجازی خود در دنبالهٔ زیر ارجاع نماید، آنگاه تعداد نقص صفحه کدام است؟

AABBCDCCEFEECD

الف) ۵ ب) ۴ ج)۳ د)۶

✓ ارشد، دولتی، ۸۴

- اگر به برنامه ای چهار صفحه اختصاص داده شود و صفحات زیر توسط برنامه به ترتیب از چپ به راست ارجاع داده شوند، تعداد پیج فالت ها چقدر خواهد بود؟

12342156212376321236

الف) با سیاست FIFO تعداد ۱۳ ب) با سیاست بهینه، تعداد ۹

ج) با سیاست LFU، تعداد ۱۰ د) هر سه مورد

nastooh@aut.ac.ir

نمونه تست های کنکور ارشد

- ارشد، دولتی، ۸۱
  - در یک سیستم مدیریتی حافظه صفحه بندی بر حسب تقاضا، اندازهٔ حافظه ۳ قاب صفحه و اندازهٔ هر قاب ۶۴ بایت است. این سیستم از الگوریتم جایگزینی LRU استفاده م یکند. یک پردازه به اندازهٔ ۹۴۰ بایت وجود دارد. با فرض اینکه هنگام اجرا به ترتیب از چپ به راست به این آدرس ها در فرآیند مراجعه شود، درصد خطای نقص صفحه چقدر است؟

40, 240, 20, 350, 160, 830, 300, 480, 800, 760, 810

الف) ۸۲٪ ب) ۷۳٪ ج) ۶۴٪ د) ۵۵٪

nastooh@aut.ac.ir

منابع

[1]. A. Silberschatz, P. B. Galvin and G. Gagne, **“Operating System Concepts,”** 9th ed., John Wiley Inc., 2013.

[2] A. S. Tanenbaum and H. Bos, “**Modern Operating Systems,**” 4rd ed., Pearson, 2014.

[3] W. Stallings, **“Operating Systems,”** 8th ed., Pearson, 2014.

[4] A. S. Tanenbaum, A. S. Woodhull**, “Operating Systems Design and Implementation,”** 3rd ed., Pearson, 2006.

[5]نستوه طاهری جوان و محسن طورانی، **“اصول و مفاهیم سیستم عامل،”** انتشارات موسسه آموزش عالی پارسه، ۱۳۸۶.

nastooh@aut.ac.ir

پایان

nastooh@aut.ac.ir