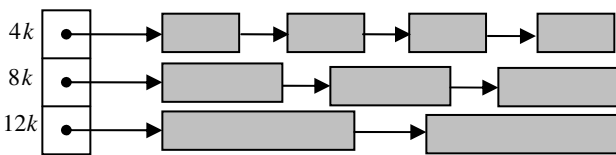


4- بدترین مناسب (worst-fit):

در این الگوریتم بزرگترین سفره انتقاب شده و پردازش در آن قرار میگیرد. دلیل انتقاب بزرگترین سفره این است که از فضای باقی مانده دیگر پردازش ها می توانند استفاده کنند. ایراد این تکنیک این است که امکان دارد، تقاضاهای که نامیه بزرگی می خواهند، دیگر نتوانند برآورده شوند چرا که بلاک های بزرگ زودتر تفصیص یافته و کوچک می شوند.

5- سریعترین مناسب (Quick Fit):

در این الگوریتم لیستی از اندازه پردازش های متداول تهیه می شود و آرایه ای با n خانه در نظر گرفته می شود که هر خانه این آرایه شامل یک اشاره گر به ابتدای لیست یک فضای خالی به اندازه متداول است به عنوان مثال فضای های متداول می توانند $2k$, $4k$, $8k$, $12k$ و ... باشند که برای هر کدام یک خانه آرایه در نظر گرفته می شود. عیب این روش این است که اگر پروسسی فائمه یابد باید فضای آزاد شده آن به لیست مناسب اضافه شود که این کار زمانبر می باشد.



6- الگوریتم رفافتی (Buddy):

در این روش سفره ها (فضاهای خالی) به صورت توان های 2 در نظر گرفته می شود. به عنوان مثال سفره هایی به اندازه $1k$, $2k$, $4k$, $8k$, $16k$, $32k$... و برای هر گروه یک لیست جداگانه در نظر گرفته می شود. بدین ترتیب جهت تفصیص یک بلاک تنها باید بلاک مورد نظر را از لیست مناسب خارج کرد. پس از تفصیص اگر فضای باقی مانده آن بلاک، توانی از 2 باشد در لیست مربوطه اش قرار می گیرد و در غیر این صورت به چندین بخش که اندازه هر کدام توانی از 2 می باشد تقسیم میشود. از طرف دیگر در این روش بلوک های کنار هم می توانند باهم ترکیب شده و بخش بزرگتری را پدید بیاورند.

مثال. با قسمت هایی از حافظه به اندازه های $100k$, $200k$, $300k$, $426k$, $500k$ هر یک از روش های اولین جای مناسب، بهترین جای مناسب و بدترین جای مناسب پردازش هایی با اندازه $100k$, $112k$, $174k$, $212k$, $200k$ را چگونه در حافظه قرار می دهند و کدام روش از حافظه به طور بهینه استفاده می کند. ترتیب ورود پردازش ها را یک بار از راست به چپ و یک بار از چپ به راست بگیرید.

H : 100 k , 500 k , 200 k , 300 , 600 k
P : 212 k , 417 , 112 k , 426 k

لیست فضای آزاد، ترتیب پردازش ها از چپ به راست
لیست فضای آزاد، ترتیب پردازش ها از راست به چپ

100 k , 176 k , 200 k , 300 k , 183 k ,
100 k , 74 k , 88 k , 88 k , 183 k

الف. اولین مناسب: در حالت از چپ به راست

پردازش 426 باید منتظر بماند

لیست فضای آزاد، ترتیب پردازش ها از چپ به راست
لیست فضای آزاد، ترتیب پردازش ها از راست به چپ

100 k , 83 k , 88 k , 88 k , 174 k ,
100 k , 74 k , 88 k , 88 k , 183 k

ب. بهترین مناسب

لیست فضای آزاد، ترتیب پردازش ها از چپ به راست
لیست فضای آزاد، ترتیب پردازش ها از راست به چپ

100 k , 388 k , 200 k , 88 k , 174 k ,
100 k , 83 k , 200 k , 188 k , 388 k

ب. بدترین مناسب

روش بهترین مناسب از حافظه به صورت بهینه استفاده می کند. زیرا برای تمام پردازش ها فضای لازم را پیدا می کند و پارگی خارجی در آن حداقل است. مثال. در زیر بلوک های خالی حافظه به ترتیب از چپ به راست نشان داده شده اند، اگر درخواست های چریدی برای چهار بلوک به اندازه $20k$, $30k$, $20k$, $35k$ به ترتیب از راست به چپ ذکر شده داده شود و از روش Next Fit استفاده شود و تفصیص از اول حافظه شروع شود، وضعیت حافظه را بعد از این تفصیص ها مشخص کنید.

شروع → 40k , 25k , 45k , 50k , 60k , 40k

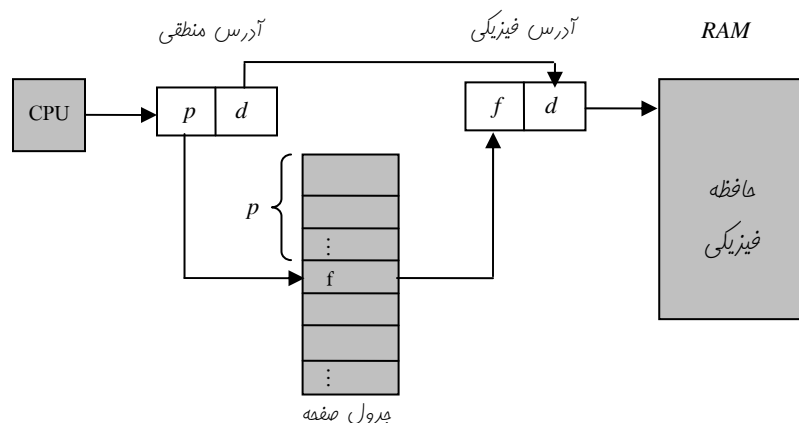
پواب: 20k , 25k , 15k , 30k , 25k , 40k

صفحه بندی (Paging):

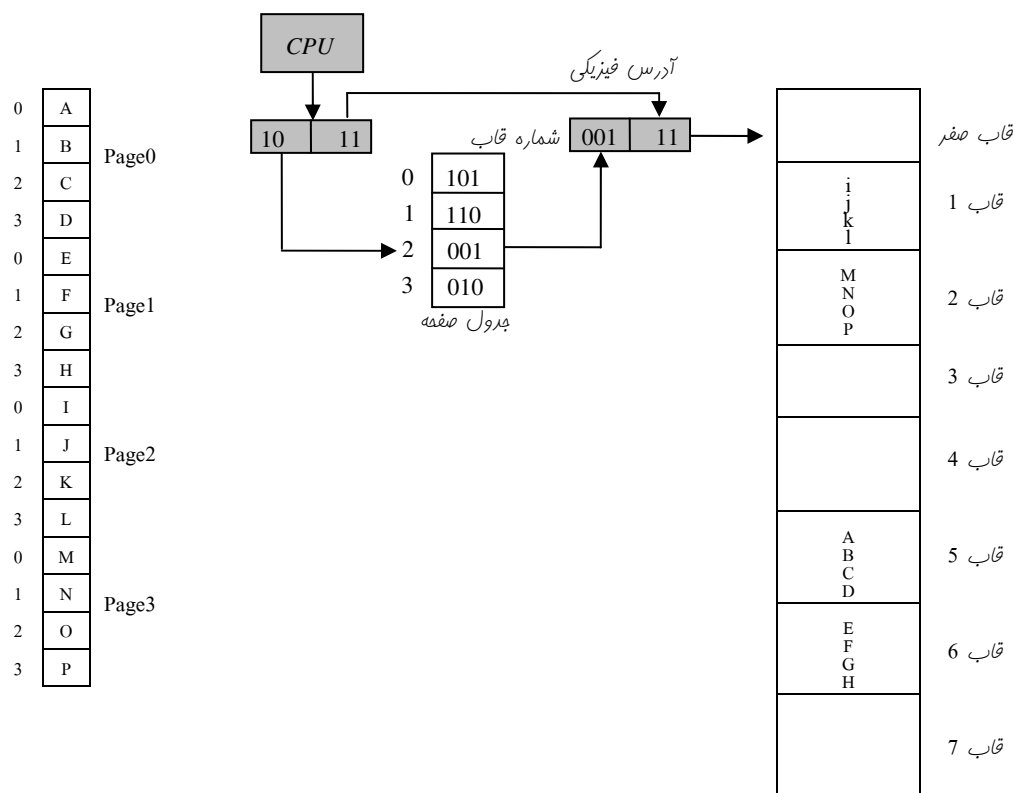
در روش صفحه بندی برنامه کاربر (فضای آدرس منطقی) به بخش هایی به اندازه ثابت به نام صفحه (Page) تقسیم می شود و حافظه فیزیکی به قسمت هایی به نام قاب (Frame) تقسیم می شود که اندازه هر فریم برابر اندازه Page می باشد. صفحه بندی این امکان را می دهد که قسمت های یک برنامه (Page های هر برنامه) در حافظه پراکنده باشند (لازم نیست مجاور هم باشند)

نمونه تبدیل آدرس منطقی به آدرس فیزیکی:

در این تکنیک هر آدرس تولید شده توسط CPU (یعنی آدرس منطقی) از دو بخش شماره صفحه (p) و افسست صفحه (d) تشکیل شده است. شماره صفحه به عنوان اندیس جدول صفحه (page table) استفاده می گردد. جدول صفحه شامل آدرس مبانی هر صفحه در آدرس فیزیکی RAM است. این آدرس مبنا با آدرس افسست منطقی ترکیب شده و آدرس فیزیکی نهائی را تشکیل می دهد. شکل زیر این موضوع را نشان می دهد.



مثال. فرض کنید تعداد صفحات $2^2 = 4$ عدد و تعداد قاب ها $2^3 = 8$ عدد باشد.



□ به هر کدام از سطرهای جدول صفحه یک مدخل یا Entry گویند. که هر مدخل شامل شماره یک فریم و اطلاعات دیگری از قبیل بیت معتبر (نامعتبر)، بیت Read، بیت write و بیت های دیگر می باشد.

$$\left[\begin{array}{c} \text{تعداد} \\ \log_2 \text{صفحات} \end{array} \right] = \text{تعداد بیت های مورد نیاز برای شماره جدول صفحه } (p)$$

حافظه منطقی \geq حافظه فیزیکی \Rightarrow تعداد page ها \geq تعداد فریم

$$\left[\begin{array}{c} \text{اندازه} \\ \log_2 \text{صفحات} \end{array} \right] = \text{تعداد بیت های مورد نیاز برای افست } (d)$$

□ در صفحه بندی اگر بیت valid صفر باشد بدین معناست که آدرس صفحه ذکر شده نامعتبر می باشد یا چنین صفحه ای وجود ندارد.

Pnu-Soal.ir