

مثال. اگر اندازه هر page ، 1k باشد تعداد خطوط آدرس منطقی را با مشخص کردن d و p و همینطور تعداد خطوط آدرس فیزیکی را با مشخص کردن d و f بدست آورید. آدرس منطقی و آدرس فیزیکی صفحه شماره 1 و خانه شماره 20 را مشخص کنید.

Page table		RAM	
00	000	000	Page0
01	111	001	
10	011	010	
11	100	011	Page2
		100	Page3
		101	
		110	
		111	Page1

حل.

p	d	F	d
10 بیت	2 بیت	10 بیت	3 بیت

$$d = \log_2^{page\ size} = \log_2^{1024} = 12$$

با توجه به شکل چهار صفحه داریم که دو بیت برای آدرس دهی آن کافی است پس برای آدرس دهی منطقی 12 بیت لازم است و

با توجه به شکل هشت قاب داریم که سه بیت برای آدرس دهی آن کافی است پس برای آدرس دهی فیزیکی 13 بیت لازم است

01	0000010100
111	0000010100

آدرس منطقی صفحه شماره 1 و خانه شماره 20

آدرس فیزیکی صفحه شماره 1 و خانه شماره 20

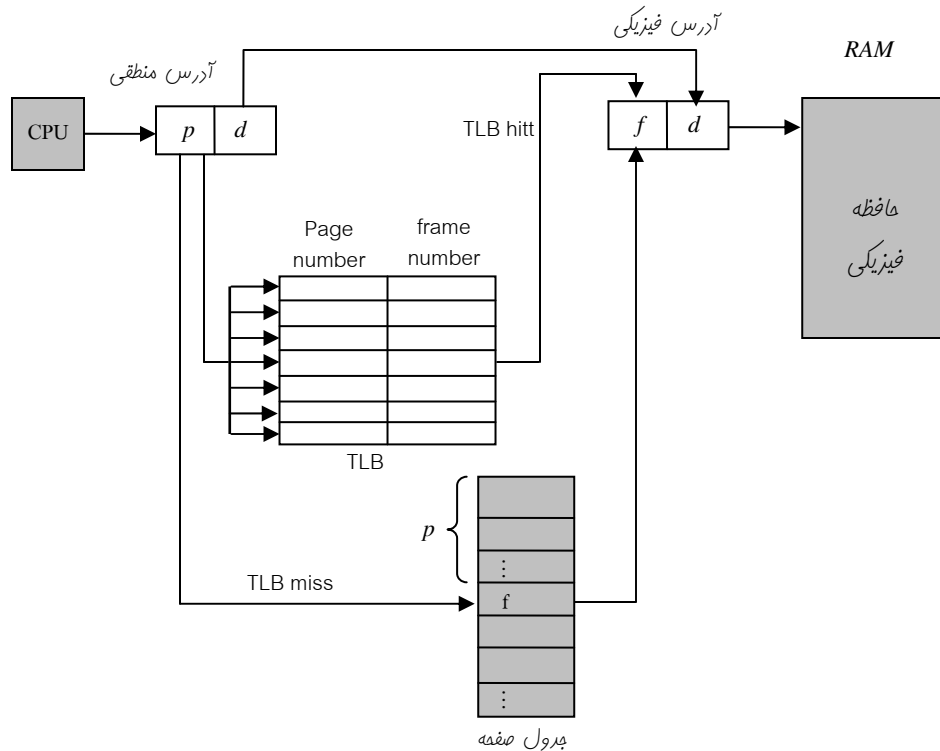
چون جدول صفحه در حافظه اصلی نگهداری میشود بنابراین هر مراجعه به حافظه تبدیل به دو مراجعه می شود. یک مراجعه به جدول صفحه جهت بدست آوردن شماره frame و مراجعه دیگر جهت بدست آوردن داده اصلی پس $t_{acc} = 2t_m$ که t_m زمان یک مراجعه می باشد. چون این روش زمانبر است معمولا جهت نگهداری جدول صفحه از بافر های دم دستی استفاده می شود (TLB) که از نوع حافظه شرکت پذیر می باشند. که به طور موازی می توان در آنها عمل جستجو را انجام داد.

روش های استفاده از TLB جهت نگهداری جدول صفحه

1- نگهداری کامل جدول صفحه در TLB

اشکال: اندازه جدول صفحه خیلی بزرگتر از TLB می باشد (اندازه TLB محدود می باشد زیرا گران است)

2- از جدول صفحه که در حافظه اصلی است به همراه TLB استفاده می شود که در این صورت جهت بدست آوردن شماره frame متناظر با یک شماره صفحه ابتدا عمل جستجو در TLB انجام میشود. اگر شماره صفحه مورد نظر در TLB یافت نشود به جدول صفحه در حافظه اصلی مراجعه میشود و بعد از این عمل TLB بروز (update) می شود. شکل صفحه بعد این موضوع را نشان میدهد.



مثال.

اگر جستجوی TLB به اندازه 20 نانو ثانیه طول بکشد و دستیابی به حافظه اصلی نیز 100 نانو ثانیه زمان بخواهد، آنگاه زمان دستیابی به حافظه را موقعی که شماره قاب در TLB پیدا شود و هنگامی که در TLB پیدا نشود را بدست آورید.

حل: زمان دستیابی اگر در TLB پیدا شود. $20 + 100 = 120 \text{ ns}$

زمان دستیابی اگر در TLB پیدا نشود. $20 + 100 + 100 = 220 \text{ ns}$

تذکر: جدول TLB در واحد مدیریت حافظه یا MMU (memory Management unit) قرار دارد، خود MMU نیز در CPU قرار دارد.

h: احتمال وجود شماره صفحه در TLB

1-h: عدم وجود (عدم موفقیت)

t_m : زمان دستیابی به حافظه اصلی

$$t_{acc} = h t_{TLB} + (1-h)(t_{TLB} + t_m) + t_m$$

مثال. یک سیستم صفحه بندی را در نظر بگیرید، جدول صفحه در حافظه اصلی است اگر زمان دستیابی به حافظه اصلی برابر 60 نانو ثانیه باشد و احتمال وجود شماره صفحه در TLB، 0.75 باشد و زمان دستیابی به TLB، 5 نانو ثانیه باشد، نسبت بهبود آدرس بر اثر TLB، در مقایسه با هنگامی که از TLB استفاده نمی شود چقدر است؟

$$h = 0.75$$

$$1-h = 0.25$$

$$t_m = 60 \text{ ns}$$

$$t_{TLB} = 5 \text{ ns}$$

زمان دستیابی به داده اصلی بدون استفاده از TLB $t_{acc} = 2t_m = 120 \text{ ns}$

زمان دستیابی به داده اصلی با استفاده از TLB $t_{acc} = 0.75(5) + (0.25)(5 + 60) + 60 = 80 \text{ ns}$

$$\text{نسبت بهبود آدرس} = \frac{120}{80}$$

مثال. در یک سیستم حافظه صفحه بندی با یک جدول صفحه حاوی 64 مدخل 11 بیتی (شامل یک بیت اعتبار/عدم اعتبار) و صفحه هایی به اندازه هر یک 512 بایت یک آدرس منطقی و یک آدرس فیزیکی چند بیت است؟

حل. چون جدول صفحه 64 مدخل و هر صفحه 512 بایت است بنابراین $2^{15} = 2^6 \times 512 = 64 \times 512$ = اندازه حافظه منطقی

پس آدرس منطقی 15 بیتی است، از طرفی چون هر مدخل جدول صفحه برای آدرس دهی 10 بیتی است (یک بیت برای عملیات کنترلی است) و هر آدرس موجود در هر مدخل (سطر) جدول صفحه به یک page با اندازه 512 بایت اشاره می کند پس $2^{19} = 2^{10} \times 512$ = اندازه حافظه فیزیکی لذا آدرس فیزیکی 19 بیتی است.

PTBR: همواره حاوی آدرس شروع جدول صفحه پردازنده در حال اجراست، هنگام context swiching، PTRB برابر آدرس شروع جدول صفحه پردازنده جدید میشود.

بعضی از مدیریت حافظه بایستی به صورت سفت افزاری باشد (پشتیبانی شود)

مثال. فرض کنید آدرس منطقی 32 بیتی و اندازه هر صفحه (قالب) نیز 1kB باشد در این صورت مطلوب است تعیین

الف. تعداد بیت های p و d

ب. اندازه جدول صفحه در صورتی که هر مدخل (Entitiy) جدول صفحه 8 بایت باشد.

حل. با توجه به این که اندازه هر صفحه یا قالب 1kB می باشد پس ($1k = 2^{10}$) 10 بیت برای قسمت آفست (d) لازم است. و با توجه به این که آدرس منطقی 32 بیتی می باشد، پس برای قسمت p، 22 بیت باقی می ماند.

p	d
---	---

10 بیت 22 بیت

برای حل قسمت ب ابتدا بایستی تعداد مدخل های جدول صفحه را بدست آوریم

$$2^{22} = 4M = \text{تعداد مدخل های جدول صفحه}$$

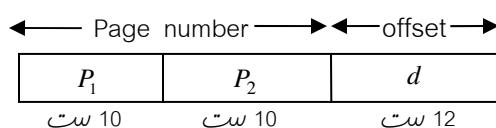
$$32M \text{ byte} = 8 \times 4M = \text{اندازه جدول صفحه}$$

□ چون اندازه جدول صفحه خیلی بزرگ می شود از روش صفحه بندی چند سطحی استفاده می شود به عبارتی برای جدول صفحه نیز جدول صفحه ایجاد می کنیم.

صفحه بندی چند سطحی

اغلب کامپیوترهای امروزی دارای فضای آدرس دهی منطقی بسیار بزرگی هستند (2^{32} تا 2^{64} فانه آدرس) در چنین سیستم هایی خود جدول بسیار بزرگ خواهد بود، برای رفع این مشکل می توان از تکنیک صفحه بندی دو سطحی استفاده کرد به گونه ای که در آن جدول صفحه، خود صفحه بندی شده باشد در واقع در این روش جدول صفحه به قطعات کوچک تقسیم شده و دیگر لازم نیست تمامی جدول صفحه در RAM نگهداری شوند به عبارت دیگر جدولی که به آنها فعلا نیاز نداریم به حافظه آورده نمی شوند

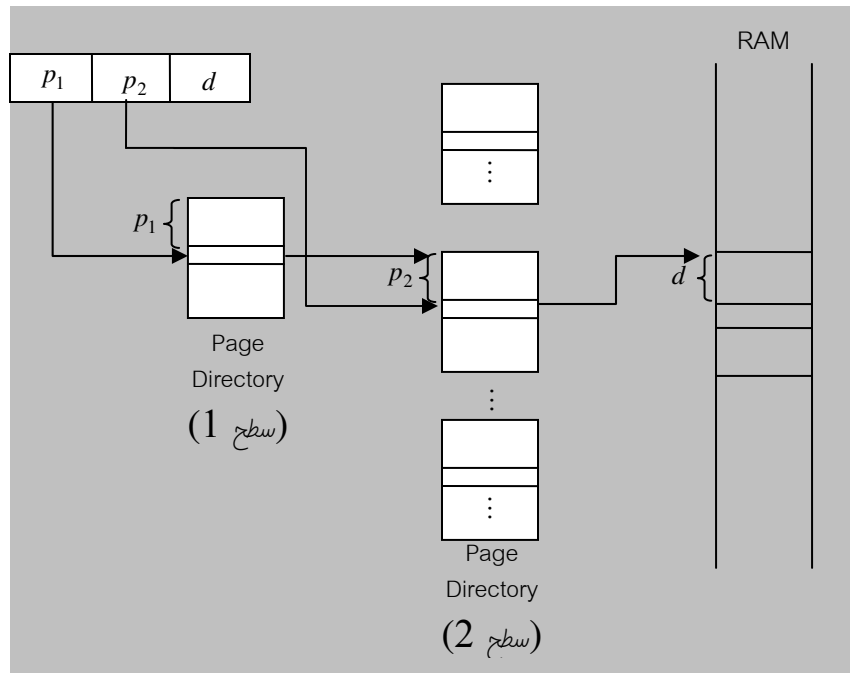
مثلا مدل آدرس دهی در اکثر پردازنده های 32 بیتی به صورت زیر است



چون آفست 12 بیتی است پس اندازه هر صفحه $2^{12} = 4k$ بوده و در اکثر

2^{20} صفحه وجود خواهد داشت ($2^{32} \div 2^{12} = 2^{20}$). هر مدخل p_1 آدرس شروع یک جدول صفحه را در سطح 2 می دهد و هر مدخل p_2 آدرس

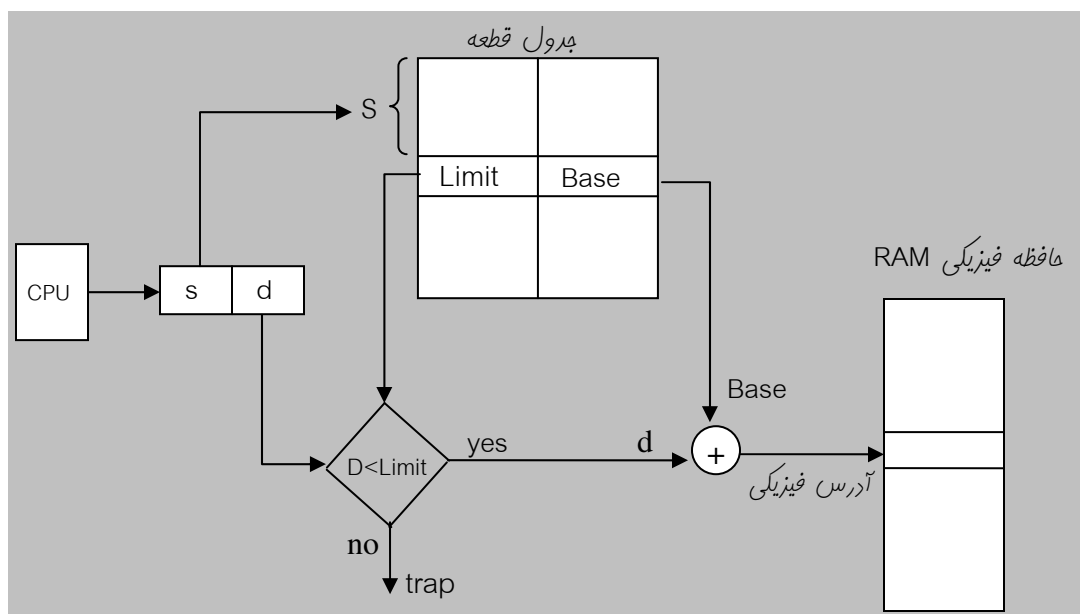
شروع یک page (یک frame) را می دهد و بیت های d یک آدرس را در داخل page (یا frame) مشخص می کند. شکل صفحه بعد نمونه این آدرس دهی را نشان می دهد.



برین ترتیب هر پردازش یک *page Directory* با $(2^{10} = 1k)$ هزار ورودی دارد که هر ورودی آن می تواند به یک *page Table* اشاره کند که آن نیز $(2^{10} = 1k)$ هزار ورودی دیگر دارد. برای هر پردازش همواره *page Directory* در حافظه قرار می گیرد ولی *page Table* های سطح 2، به تعداد لازم در حافظه قرار می گیرند.

قطعه بندی (segmentation)

حافظه اصلی یا فیزیکی به صورت یک آرایه فطری از بایت ها می باشد به عبارتی هر خانه در حافظه یک آدرس دارد، در روش صفحه بندی پردازش ها بر اساس محدودیت فیزیکی یعنی اندازه هر صفحه تقسیم می شوند ولی در قطعه بندی خود کاربر برنامه نویس می تواند برنامه اش را به صورت منطقی تقسیم کند که به هر یک از این قسمت ها یک *segment* گویند هر قطعه یک آدرس شروع و یک طول دارد، آدرسی که کاربر در سطح منطقی می دهد شامل دو جزء است. یکی شماره قطعه و دیگری فاصله (آفست) خانه مورد نظر از اول آن قطعه است، آدرس دو بعدی استفاده شده توسط کاربر در سطح منطقی می بایست توسط یک نگاشت به آدرس یک بعدی فیزیکی تبدیل شود. این نگاشت به وسیله جدول قطعه انجام می پذیرد، جدول قطعه از دو ستون اصلی تشکیل شده است، یکی آدرس پایه قطعه (Base) و دیگری طول یا حد قطعه (Limit). قسمت Base حاوی آدرس فیزیکی در RAM می باشد که قطعه از آنجا شروع می شود، شکل زیر نحوه تبدیل آدرس منطقی را به آدرس فیزیکی در این سیستم نشان میدهد.



آدرس منطقی از دو جزء شماره قطعه (s) و آفست درون آن قطعه (d) تشکیل یافته است.

□ یکی از مزایای قطعه بندی اشتراک است.

□ در صفه بندی اندازه صفحات برابر است ولی در قطعه بندی لزومی ندارد که اندازه صفحات یکی باشد

مثال. فرض کنید در یک شبکه کامپیوتری چندین کاربر به صورت همزمان نیاز به اجرای برنامه word دارند به جای این که هر برنامه word که برای همه مشترک است برای هر کاربر به صورت مجزا در حافظه load شود، فقط یک بار در حافظه load شده و تمامی کاربران به صورت اشتراکی از آن استفاده می کنند و هر کاربر قطعه خاص خود را دارد.

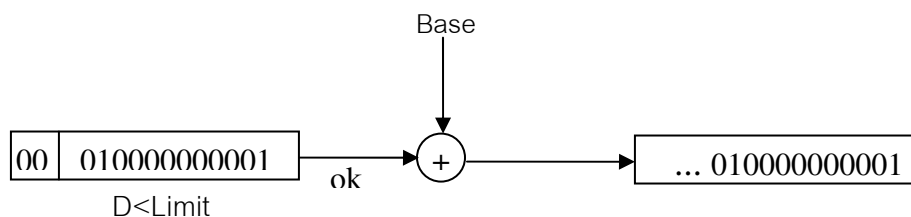
مثال. فرض کنید پردازش ای دارای سه قطعه می باشد به طوریکه قطعه اول 2k، قطعه دوم 3k و قطعه سوم 4k می باشد مطلوب است الف. تعیین ورودیهای جدول قطعه

	base	Limit	V
00	0000	2×2^{10}	1
01	3000	3×2^{10}	1
10	2000	4×2^{10}	1
11			0

	RAM
0000	قطعه ی 0
2000	قطعه ی 2
3000	قطعه ی 1

S	d
2	12

ب. اگر برنامه نویس در قطعه صفر دستور push با آدرس 1025 را داشته باشد آدرس فیزیکی متناظر با این آدرس را بیابید.



نکته: اگر برنامه نویس دستور pop فانه 2048 را در قطعه ی صفر بدهد چون آدرس صادر شده در قطعه صفر وجود ندارد بنابر این وقفه رخ میدهد

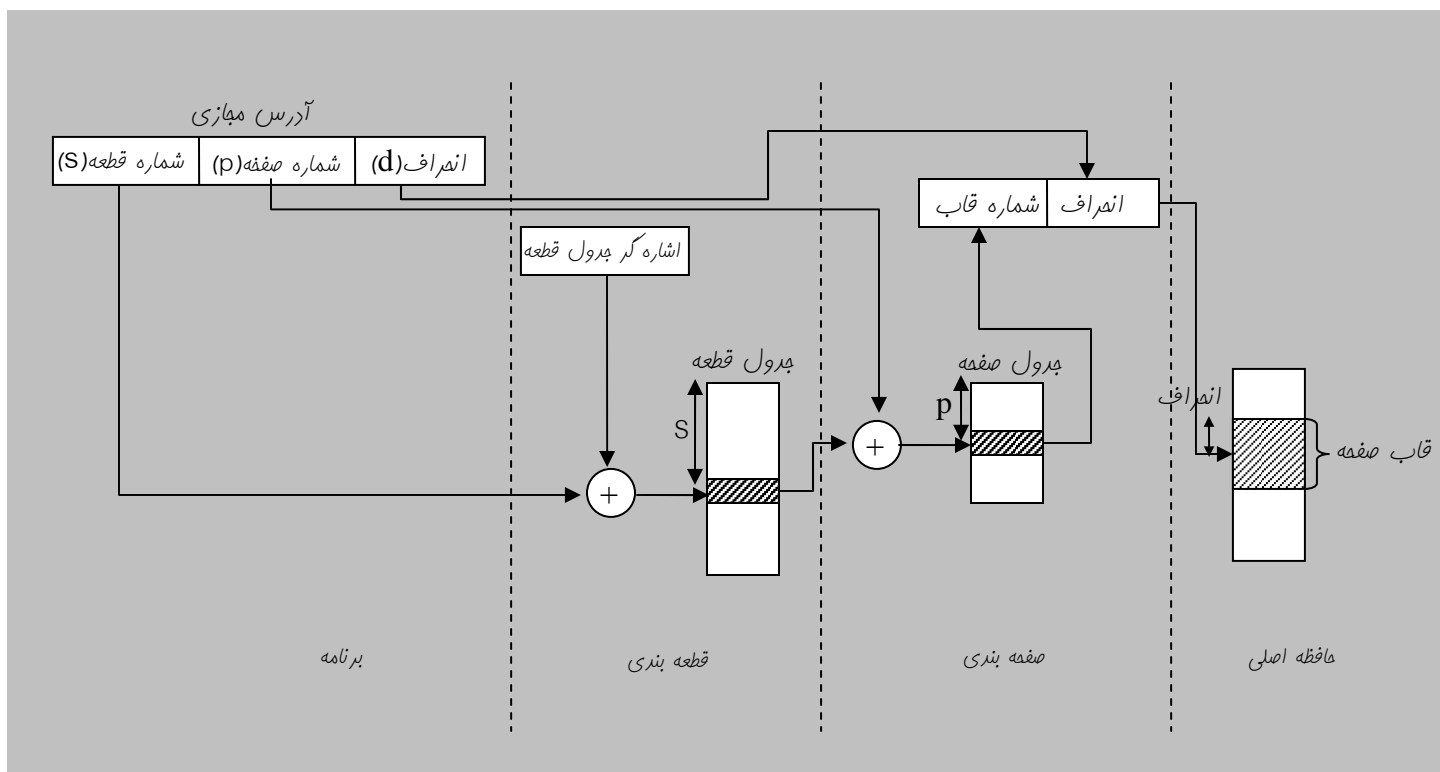
ج. اگر برنامه نویس دستور `Mov AX, [100]` را صادر کند، آدرس موثر مطلق (فیزیکی) را بیابید

ترکیب قطعه بندی و صفه بندی

هم صفه بندی و هم قطعه بندی نقاط قوت مخصوص به خود را دارند برای ترکیب نقاط قوت هر دو، این دو روش را ترکیب کرده و با هم استفاده می کنیم، در یک سیستم ترکیبی صفه بندی / قطعه بندی فضای آدرس کاربر تحت نظر برنامه ساز به تعدادی قطعه تقسیم میشود، هر قطعه به نوبه خود به تعدادی صفه به اندازه ثابت تقسیم می گردد. به ازای هر فرایند یک جدول قطعه و به ازای هر قطعه یک جدول صفه ایجاد می گردد، تعداد درایه های هر جدول صفه بستگی به اندازه قطعه مربوطه دارد. همانند صفه بندی آخرین صفه هر قطعه معمولاً پر نمی شود و به ازای هر قطعه به طور میانگین به اندازه نصف صفه تکه تکه شدن داخلی داریم. از دید برنامه ساز، آدرس

منطقی همپنان شامل شماره قطعه و انصراف در قطعه است. از دید سیستم این انصراف در قطعه، به صورت یک شماره صفحه و انصراف در صفحه دیده می شود.

هنگامی که فرایندی در حال اجراست یک ثابت آدرس شروع جدول قطعه آن فرایند را نگه می دارد. پردازنده از شماره قطعه ای که در آدرس منطقی است به عنوان شاخص به جدول قطعه استفاده کرده و جدول صفحه را برای قطعه مذکور پیدا می کند، سپس بخش شماره صفحه آدرس منطقی به عنوان شاخص جدول صفحه به کار رفته و شماره قاب مربوطه بدست می آید. این شماره قاب با بخش انصراف آدرس منطقی ترکیب شده و آدرس فیزیکی مورد نظر نتیجه میشود. شکل زیر نحوه ترجمه آدرس در یک سیستم قطعه بندی/صفحه بندی را نشان می دهد



برای مثال سیستم با آدرس منطقی 34 بیتی را در نظر بگیرید که در آن شماره قطعه 18 بیتی و انصراف 16 بیتی است، به کارگیری مدیریت حافظه قطعه بندی در این سیستم به خاطر اندازه بزرگ قطعه که می تواند تا 2^{16} باشد مشکلاتی را که در فوق متذکر شدیم در بر می گیرد. بنابراین سیستم از مدیریت حافظه قطعه بندی/صفحه بندی استفاده می نماید. نگارش این مدیریت در این سیستم بدین گونه است که انصراف قطعه به دو بخش 6 بیتی شماره صفحه و 10 بیتی انصراف صفحه تجزیه می شود. جدول صفحه برای هر قطعه می تواند حداکثر از 2^6 درایه برخوردار باشد و هر فرایند حداکثر می تواند تا 2^{18} قطعه را دربرگیرد.

انصراف از صفحه (10 بیت)	شماره صفحه (6 بیت)	شماره قطعه (18 بیت)
-------------------------	--------------------	---------------------

مهمترین مزیت صفحه بندی:

پارگی فاربی ندارد

در قطعه بندی پارگی فاربی داریم زیرا اندازه قطعات مساوی نیست و قطعه ای بلااستفاده ای وجود دارد که پراکنده بودن و همجواری نیستند بنابراین نمی توان از آنها برای یک قطعه استفاده کرد.

مهمترین مزیت قطعه بندی

اشتراک

اشتراک در صفحه بندی نسبت به قطعه بندی ناپیژ است و یا اصلا وجود ندارد.

جدول زیر مقایسه ای بین روش های مدیریت حافظه را نشان می دهد.

آیا کل برنامه در حافظه اصلی کنار هم قرار می گیرد	جهت اجرا کل برنامه در حافظه می باشد یا نه	چند برنامه	
کنار هم	کل برنامه	یک	تک برنامه کی ساده
کنار هم	لزومی ندارد	یک	تک برنامه کی با overlay
کنار هم	کل برنامه	چند	چند برنامه کی با swapping
کنار هم	کل برنامه	چند	چند برنامه کی به صورت همجواری
کنار هم	کل برنامه	چند	چند برنامه کی به صورت بخش بندی
لزومی ندارد	کل برنامه	چند	صفحه بندی
لزومی ندارد	کل برنامه	چند	قطعه بندی

Pnu-Soal.ir