

حافظه مجازی:

تکنیکی است که اجازه می دهد بدون این که کل برنامه در حافظه اصلی قرار گیرد اجرا شود. **مزیت مهم** این روش آن است که اجازه می دهد برنامه بزرگتر از حافظه اصلی باشد. علاوه بر این در این روش کاربرد حافظه را به صورت آرایه ای فوق العاده بزرگ و یکنواخت می بیند. به عبارتی کاربرد (برنامه نویسنده) خود را درگیر حافظه فیزیکی نمی کند و فرض می کند که بی نهایت حافظه در اختیار دارد و برنامه نویسی ساده می شود. **مزیت دیگر** این است که چون فقط صفات مورد نیاز از دیسک به حافظه اصلی منتقل می شوند بنابراین زمان کمتری صرف عمل ۱/۰ می شود از این رو زمان پردازش نیز سریعتر می شود.

بهرت پیاده سازی حافظه مجازی دو روش وجود دارد.

- ترکیبی از صفحه بندی و swapping

- ترکیبی از قطعه بندی و swapping

به عبارتی حافظه مجازی به صورت صفحه بندی نیازی پیاده سازی می شود. در این روش مطابق اصل مملی بودن مراجعه به حافظه صفحه یا صفحاتی را که مورد نیاز باشند به حافظه اصلی آورده می شوند و هر وقت به صفحه ای نیاز باشد ولی در حافظه اصلی نباشد آن صفحه مطابق روشی که بعدا خواهیم گفت به حافظه اصلی Load می شود.

حافظه فیزیکی به یک سری قسمت های مساوی به نام Frame یا قاب تقسیم می شود. و برنامه کاربرد نیز به قسمت هایی به اندازه هر Frame به نام page یا صفحه تقسیم می شود.

□ اگر در ابتدا هیچ صفحه ای در حافظه اصلی نباشد و به ممض نیاز، به حافظه اصلی آورده شود به این روش **صفحه بندی نیازی ممض** گویند. در این روش هرگز صفحه ای تا وقتی که مورد نیاز نباشد وارد حافظه نمی شود.

کارایی صفحه بندی نیازی

اگر t_m زمان دستیابی به حافظه، p احتمال وقوع فضای صفحه (احتمال این که صفحه در حافظه اصلی نباشد) و t_f زمان لازم برای سرویس دهی به فضای صفحه باشد، آنگاه زمان دسترسی موثر (t_{eff}) از فرمول زیر مناسبه می شود.

$$t_{eff} = (1 - p)t_m + p \times t_f$$

در اکثر سیستم ها t_m بین 10 تا 200 نانو ثانیه می باشد.

انتساب قاب ها یا frame ها به پردازه ها

- **انتساب مساوی.** در این روش قاب های حافظه اصلی بین پردازه ها به صورت مساوی تقسیم می شود به عنوان مثال اگر حافظه اصلی دارای 12 قاب باشد و 5 پردازه بخواهند به طور هم زمان اجرا شوند به هر کدام از پردازه ها دو قاب تفصیص می یابد و دو قاب باقی مانده به عنوان قاب های بافر در نظر گرفته می شود. پس اگر n قاب داشته باشیم معمولا k تا از این n قاب را به عنوان بافر در نظر گرفته و $n-k$ قاب باقی مانده را مابین m پردازه تقسیم می کنیم.

- **انتساب متناسب.** در این روش تعدا قاب ها متناسب با اندازه پردازه ها به پردازه ها تفصیص می یابد به عنوان مثال اگر حافظه اصلی دارای 12 قاب باشد و دو قاب از آنها به عنوان بافر در نظر گرفته شود در صورتی که اندازه هر قاب $1k$ باشد و در

سیستم سه پرده 2k, 3k, 5k به طور همزمان در حال اجرا باشند به پرده 5k, 5 پرده 3k, 3 پرده 2k و به پرده 2k, 2 پرده 2k, 2 پرده 2k در نظر گرفته می شود.

هیچ کدام از دو روش قبلی الویت را در نظر نمی گیرند می توان کاری کرد که به پرده ها به نسبت الویتشان قاب تفصیل داد.

در هر مراجعه به حافظه اعمال زیر رخ می دهد

در روش مدیریت حافظه مجازی به آدرس منطقی آدرس مجازی می گویند اگر بیت ۷ یک باشد بدین معناست که page مورد نیاز در حافظه اصلی موجود است اگر صفر باشد دو برداشت از آن وجود دارد.

1- پنین page ی اصلا وجود ندارد یعنی دسترسی غیر مجاز است به عنوان مثال پرده شما دارای سه page است ولی شما مراجعه به page چهارم را می دهید که این دسترسی غیر مجاز است.

2- پنین page ی وجود دارد ولی در حافظه اصلی نیست.

در روش حافظه مجازی به طریق صفحه بندی نیازی، در هر مراجعه به حافظه مراحل زیر دنبال می شود.

1- ابتدا به ساختار جدول صفحه مراجعه می شود اگر صفحه مورد درخواست در حافظه اصلی باشد شماره قاب از ساختار جدول صفحه درست آمده بنابراین آدرس فیزیکی آماده می شود.

2- اگر بیت $v=0$ باشد و صفحه مراجعه شده در پرده وجود نداشته باشد، دسترسی غیر مجاز است و پایان پرده است

3- اگر بیت $v=0$ باشد ولی صفحه مورد نظر جزء پرده باشد در این صورت خطای فقدان صفحه رخ داده و بنابراین

الف. در حافظه اصلی قاب خالی وجود داشته باشد، صفحه مورد نظر از رسانه جانبی به حافظه اصلی منتقل شده و جدول صفحه update می شود.

ب. در حافظه اصلی قاب خالی وجود ندارد از این رو یکی از صفحات موجود در حافظه اصلی مطابق الگوریتم های جایگزینی به رسانه جانبی منتقل شده و صفحه مورد نیاز از رسانه جانبی به حافظه اصلی منتقل می شود.

برای این که انتقال صفحه مورد نیاز از حافظه جانبی به حافظه اصلی منتظر خالی شدن قاب حافظه اصلی نباشد، صفحه مورد نیاز از حافظه جانبی به یکی از قاب های بافر منتقل شده و بنابراین عمل ورودی و خروجی سریعتر انجام میشود. سپس بعد از انتقال قاب شایسته ی خروج از حافظه اصلی به حافظه جانبی، قاب خالی شده به لیست قاب های بافر اضافه می شود.

الگوریتم های جایگزینی صفحه

مساله الگوریتم های جایگزینی صفحه یعنی هنگامی که نیاز است صفحه ای را از حافظه خارج کنیم کدام یک را انتخاب کنیم تا کارایی حداکثر شود. در زیر الگوریتم های متعددی را برای جایگزینی صفحه شرح می دهیم. ولی به طور کلی الگوریتمی بهتر است که تعداد نقض صفحه های (page fault rate) آن کمتر باشد بدیهی است که هر چه قدر تعداد فریم های آزاد بیشتر باشد، تعداد خطای صفحه ای کاهش می یابد.

1- الگوریتم FIFO : ساده ترین روش جایگزینی صفحه است. در این روش صفحه ای برای خروج انتخاب می شود که زودتر از بقیه صفات وارد حافظه اصلی شده باشد

مثال. مراجعات (از چپ به راست) 0, 1, 2, 3, 0, 1, 4, 0, 1, 2, 3, 4 را با سه قاب آزاد در نظر بگیرید در روش جایگزینی FIFO چند نقض صفحه رخ می دهد.

تقاضا	0	1	2	3	0	1	4	0	1	2	3	4
قاب 1	0	0	0	3	3	3	4	4	4	4	4	4
قاب 2		1	1	1	0	0	0	0	0	2	2	2
قاب 3			2	2	2	1	1	1	1	1	3	3
نقض صفحه	*	*	*	*	*	*	*			*	*	

ملاحظه می شود که در این مثال 9 بار فضای نقض صفحه رخ می دهد.

ناهنجاری بلیدی: اگر در الگوریتم جایگزینی با افزایش تعداد قاب ها (Frame) تعداد فضای صفحه افزایش یابد کوئیم آن الگوریتم دارای ناهنجاری بلیدی است که الگوریتم FIFO یکی از آنها می باشد.

ادامه دارد.....

Pnu-Soal.ir