



دانشکده علوم ریاضی و آمار



نیمسال دوم ۱۴۰۱-۱۴۰۰

مدرس: دکتر مجتبی رفیعی

## اصول سیستم‌های عامل

### جلسه ۱۲

نگارنده: محمد مهدی عباسی

۸ فروردین ۱۴۰۱

## فهرست مطالب

- |   |                                      |
|---|--------------------------------------|
| ۱ | مطالب تکمیلی- عمل Mount کردن         |
| ۲ | مدیریت حافظه‌ی نهان                  |
| ۳ | مدیریت سیستم I/O                     |
| ۳ | فرآیندها (Processes)                 |
| ۴ | ۱.۴ چیدمان حافظه یک فرآیند . . . . . |

## ۱ مطالب تکمیلی- عمل Mount کردن

فرآیندی است که در آن سیستم عامل، فایل‌ها و پوشه‌های روی یک رسانه ذخیره‌سازی را برای کاربران قابل دسترسی می‌کند. برخی از عملیات‌هایی که در این فرآیند بررسی و راه‌اندازی می‌شود عبارتند از:

۱. شناسایی رسانه ذخیره‌سازی،
۲. پردازش ساختار فایل و فراداده‌های روی آن،
۳. خواندن از رسانه،

۴. نوشتن بر روی رسانه.

unmounting فرآیند عکس mounting است که در آن سیستم عامل، تمام دسترسی‌های کاربران به فایل‌ها و پوشه‌های موجود در نقطه mount شده را قطع می‌کند، صف داده‌های کاربر را روی رسانه می‌نویسد (داده‌هایی که ثبت نهایی نشده‌اند)، فراداده‌های سیستم فایل را تازه سازی (Refresh) می‌کند و دستگاه ذخیره‌سازی را برای جدا شدن ایمن از سیستم کامپیوتری آماده می‌کند.

## ۲ مدیریت حافظه‌ی نهان

عمل caching یک اصل مهم در سیستم‌های کامپیوتری است که در آن داده‌ها از یک حافظه ذخیره‌سازی کندتر به‌طور موقت به حافظه سریعتر کپی می‌شوند. به عنوان مثال، داده‌ها از حافظه اصلی به کش کپی می‌شوند. به هنگام پردازش، اگر داده در کش باشد به طور مستقیم از آن استفاده می‌شود و اگر چنین نبود از حافظه اصلی یک کپی از آن را به کش انتقال می‌دهیم و با این فرض که ممکن است این داده به زودی توسط پردازشگر مورد استفاده قرار گیرد، آن را در کش نگه می‌داریم.

اگر حافظه کش به درستی مدیریت نشود: cpu برای انجام پردازش می‌بایست چند چرخه اجرایی برای واکنشی داده از حافظه اصلی منتظر بماند. از طرفی نیز حجم حافظه کش محدود بوده و این خود بر چالش مدیریتی آن می‌افزاید. پارامترهای زیر برای مدیریت حافظه کش به منظور افزایش کارایی یک سیستم کامپیوتری دارای اهمیت است:

۱. انتخاب دقیق اندازه حافظه نهان،

۲. سیاست جایگزینی (Policy Replacement).

Level	1	2	3	4	5
Name	registers	cache	main memory	solid-state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25-0.5	0.5-25	80-250	25,000-50,000	5,000,000
Bandwidth (MB/sec)	20,000-100,000	5,000-10,000	1,000-5,000	500	20-150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

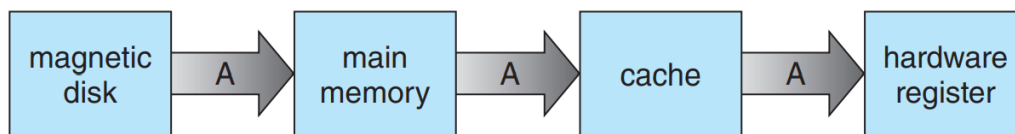
شکل ۱: ویژگی‌های انواع حافظه‌های ذخیره‌سازی

مطلب تکمیلی. ، انتقال داده بین سطوح مختلف از سلسله مراتب حافظه می‌تواند به صورت مستقیم یا غیرمستقیم انجام شود و این امر وابسته به طراحی سخت‌افزار و سیستم عامل است. به عنوان مثال، انتقال داده از حافظه کش به cpu و ثبات‌ها به طور معمول از طریق یک تابع سخت افزاری و بدون مداخله سیستم عامل انجام می‌شود، درحالی‌که انتقال داده از دیسک به حافظه اصلی به طور معمول به وسیله سیستم عامل کنترل می‌شود.

با توجه به مباحث قبلی، یک داده یکسان می‌تواند در سطوح مختلف سلسله مراتب حافظه در یک سیستم کامپیوتری وجود داشته باشد. برای فهم بهتر مطلب و چالش‌های داشتن چند کپی یکسان در سطوح مختلف، مثال زیر را در نظر بگیرید. فرض کنید عدد صحیح A در فایل B که مستقر در HDD است، قرار دارد و می‌خواهیم یک واحد به آن اضافه کنیم. عمل افزایش به صورت زیر انجام می‌شود:

۱. ابتدا یک عملیات I/O برای کپی A به حافظه اصلی انجام می‌شود،

۲. سپس یک کپی از حافظه اصلی به کش و به دنبال آن کپی در رجیسترهای داخلی CPU انجام می‌شود. در این گام کپی‌های یکسان از A در سطوح مختلف حافظه (HDD, Main Memory, cache, Register) وجود دارد.



شکل ۲: انتقال عدد صحیح A از دیسک به Register

۳. هنگامی که افزایش مقدار A در رجیستر اتفاق می‌افتد، مقدار A متفاوت از سطوح مختلف می‌شود. در حالت کلی، از دید ما زمانی یکسان می‌شود که مقدار A از رجیستر برگردانده شده و روی HDD نوشته شود.

با این حال، فرآیندهای در حال اجرا برای دسترسی به یک مقدار یکسان از داده با چالش‌هایی روبرو هستند:

۱. در محیط چند وظیفه‌ای: دسترسی به یک مقدار معتبر و یکسان توسط سایر فرآیندها ضروری است،
۲. در محیط چند پردازنده‌ای: شرایط حادث می‌شود چراکه در یک سیستم کامپیوتری شامل چند هسته پردازشی با کش‌های داخلی، حافظه‌های محلی و ... هستند.
۳. در محیط توزیع شده: نسبت به دو محیط قبل، شرایط حادثتری داریم چراکه به عنوان مثال یک فایل روی چند سیستم کامپیوتری مستقل کپی شده‌اند.

### ۳ مدیریت سیستم I/O

یکی از اهداف سیستم عامل، پنهان کردن ویژگی‌های دستگاه‌های سخت افزاری از کاربران است. به طور معمول این وظیفه در سیستم عامل بر عهده زیر سیستم I/O (Subsystem I/O) است. لازم به ذکر است که تنها Device Driver ها (و نه کاربران عادی سیستم) در چنین رویکردی از ویژگی‌های دستگاه‌های I/O آگاه هستند.

زیر سیستم I/O (I/O subsystem) شامل چندین مولفه به قرار زیر است:

۱. مولفه مدیریت حافظه که شامل

- بافر کردن (Buffering): ذخیره سازی موقت داده حین انتقال،
- کش کردن (Caching): ذخیره کردن بخشی از داده در رسانه ذخیره‌سازی سریعتر برای افزایش کارایی،
- اسپولینگ (Spooling): هم‌پوشانی خروجی یک کار با ورودی کارهای دیگر

۲. رابط عمومی Device driver ها

۳. درایورهایی برای دستگاه‌های سخت افزاری خاص

### ۴ فرآیندها (Processes)

به طور غیر رسمی یک برنامه در حال اجراست و به عنوان واحد کار در یک سیستم کامپیوتری شناخته می‌شود. در کامپیوترهای نسل قدیم، از آنجایی که تمهیدات لازم تنها برای اجرای یک برنامه در سیستم کامپیوتری در نظر گرفته می‌شد، یک برنامه کنترل کامل سیستم و در نتیجه دسترسی به همه منابع سیستم را داشت. بنابراین چالش‌های مدیریتی به مراتب کمتری نسبت به کامپیوترهای امروزی که اجرای هم‌راند چندین برنامه را مسیر ساخته‌اند، وجود داشت.

سیستم‌های امروزی از منظر فرآیند، شامل گردایه‌ای از فرآیندهای کاربر (کدهای کاربر را اجرا می‌کنند) و فرآیندهای سیستمی (که کدهای سیستم عامل را اجرا می‌کنند) است که در میان یکدیگر اجرا می‌شوند.

سوال: فعالیت‌های cpu را چه بنامیم؟

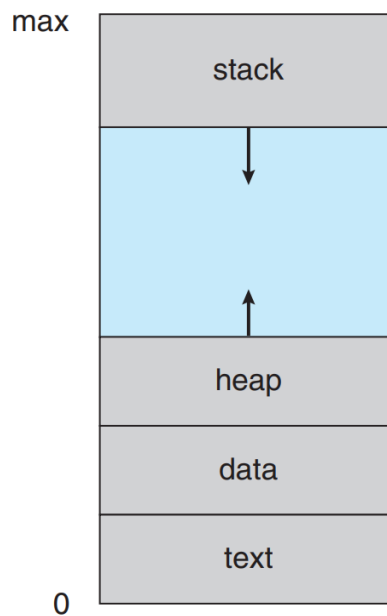
پاسخ: از منظر تاریخی و از آنجایی که وظیفه اصلی cpu پردازش job های کاربران بوده است، اصطلاح job (کار) برای آن در نظر گرفته شده است. با این حال اصطلاح امروزی مورد استفاده برای فرآیند Process تغییر کرده است.

## ۱.۴ چیدمان حافظه یک فرآیند

همانطور که دیدیم، فرآیندها قبل از تصاحب پردازنده برای اجرا می‌بایست در حافظه اصلی بارگذاری شود. چیدمان حافظه برای یک فرآیند، شامل بخش‌های زیر است:

- بخش متن (Text Section): حاوی کد اجرایی برنامه است.
- بخش داده (Data Section): حاوی متغیرهای عمومی برنامه است.
- بخش هیپ (Heap Section): حافظه‌ای که به‌طور پویا در حین اجرای برنامه به آن تخصیص داده می‌شود.
- بخش پشته (Stack Section): حافظه موقت که به هنگام فراخوانی توابع مورد استفاده قرار می‌گیرد و حاوی متغیرهای محلی، آدرس بازگشت، پارامترهای تابع و... است.

شکل زیر چیدمان یک فرآیند در حافظه اصلی را نشان می‌دهد.



شکل ۳: چیدمان یک فرآیند در حافظه اصلی

معرفی برخی نکات در رابطه با چیدمان حافظه یک فرآیند:

۱. اندازه بخش Text و بخش Data در طول اجرای برنامه ثابت است، با این حال بخش Stack و Heap به‌طور پویا می‌توانند در طول اجرای برنامه کوچک شده یا رشد پیدا کنند.
۲. بخش Heap و بخش Stack به سمت یکدیگر رشد پیدا می‌کنند و سیستم عامل می‌بایست اطمینان عدم هم‌پوشانی این دو را تضمین کند.
۳. هنگامی که در سیستم یک تابع فراخوانی می‌شود، یک رکورد فعالیت یا Activation Record حاوی اطلاعات زیر تولید و به پشته اضافه می‌شود:

- پارامترهای تابع،
- متغیرهای محلی،
- و آدرس بازگشت.

در نهایت نیز به هنگام بازگشت از تابع، رکورد فعالیت از بالای پشته حذف می‌شود. بنابراین رشد و کاهش حافظه مصرفی مربوط به پشته به صورت مذکور انجام می‌شود.

به طور مشابه، هنگامی که برنامه به صورت پویا درخواست تخصیص حافظه می‌کند، منجر به رشد حافظه مصرفی بخش Heap می‌شود و به هنگام رهاسازی حافظه مصرفی، حافظه Heap آزاد شده و مقدار اشغال شده این حافظه کاهش می‌یابد.

مثال چیدمان حافظه. شکل زیر چیدمان حافظه برای یک نمونه برنامه زبان C را نشان می‌دهد.

