

سیستم های عامل

رشته مهندسی کامپیوتر

ناصر آیت

عناوین فصل ها:

فصل اول : نگاه کلی به سخت افزار.

فصل دوم: نگاه کلی به سیستم عامل.

فصل سوم: فرآیندها.

فصل چهارم: نخها، چند پردازشی متقارن و زیر هسته ها

فصل پنجم: همزمانی: انحصار متقابل و همگام سازی.

فصل ششم : بن بست و گرسنگی.

فصل هفتم: مدیریت حافظه.

فصل هشتم: حافظه مجازی.

فصل نهم: زمان بندی تک پردازنده ای

فصل اول

نگاه کلی به سخت افزار

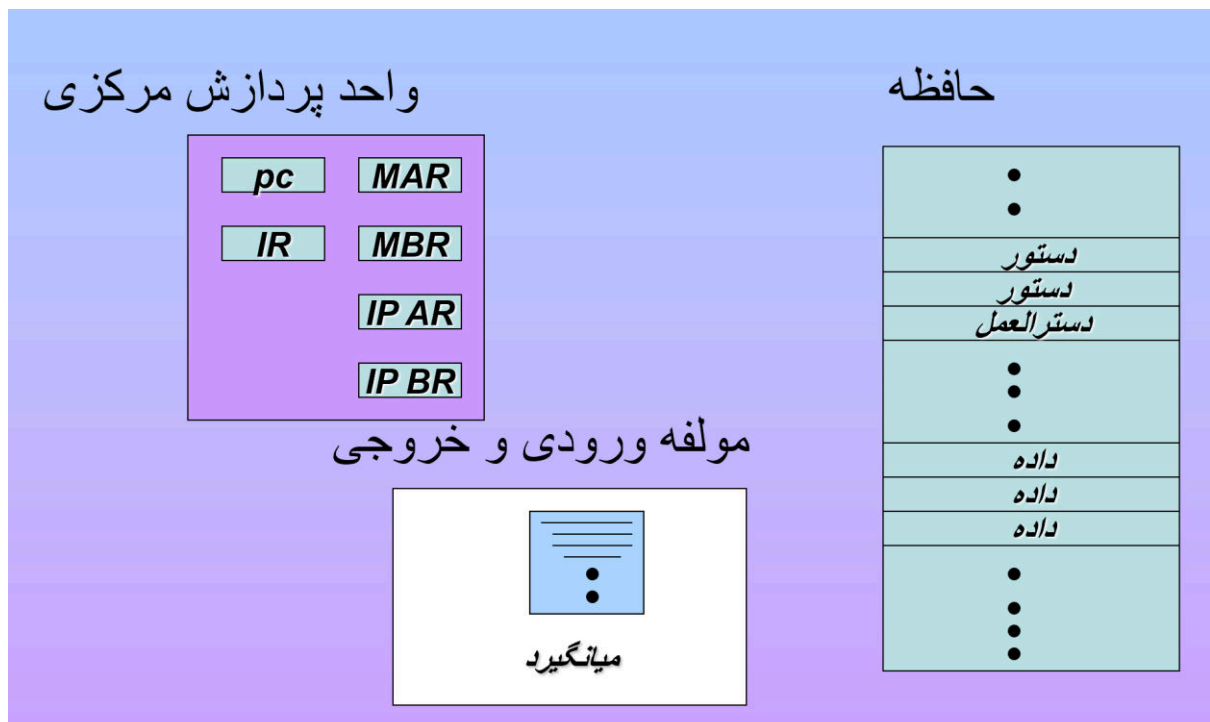
نگاه کلی به سخت افزار

چهار عنصر اصلی سخت افزار:

- ۱- پردازنده (cpu): کنترل و پردازش داده ها .
- ۲- حافظه اصلی: ذخیره داده ها و برنامه ها و نا پایدار.
- ۳- مولفه های ورودی و خروجی: انتقال داده ها بین کامپیوتر و محیط خارجی .
- ۴- اتصالات داخلی سیستم: جهت ارتباط بین سه مولفه دیگر .

واحد پردازش مرکزی control processing unit

حافظه



واحد پردازش مرکزی

- PC : حاوی دستورالعملی که باید واکنشی شود .

- IR : ثبات دستور العمل حاوی آخرین دستور العمل واکشی شده .
- MIR : ثبات آدرس حافظه، محل خواندن و نوشتن را مشخص می کند .
- MBR: ثبات میانگیر حافظه، در برگیرنده داده ای است که قرار است خوانده یا نوشته شود.
- I/O AR : ثبات آدرس ورودی خروجی، مشخص کردن یک دستگاه ورودی یا خروجی خاص .
- I/O BR : ثبات میانگیر ورودی خروجی، برای تبادل داده بین پردازنده و مولفه ورودی خروجی .

ثباتهای پردازنده

حافظه سریعتر و کوچکتر از حافظه اصلی است که در داخل پردازنده قرار گرفته است

دو وظیفه آن:

۱- مراجعه به حافظه اصلی را به حداقل می رساند.

(قابل رویت هستند)

۲- کنترل عملیات پردازنده

(ثبات کنترل و وضعیت ؛ اغلب قابل رویت نیستند)

حافظه :

شامل مجموعه ای از محل هایی است ، که حاوی یک عدد دودویی است ، که می توانیم دستورالعمل یا داده تفسیر شود که بوسیله شماره آدرس هایی برای آنها مشخص می شود.

مولفه ای ورودی / خروجی

داده ای میانگیر داخلی است.

جهت نگهداری داده ها تا زمان انتقال

ثبات قابل رویت برای کاربرد

ثباتی است که برنامه نویس می تواند به وسیله دستور العمل های ماشین به آنها مراجعه کند.

- ۱- **ثباتهای داده:** برنامه ساز می تواند به بعضی توابع نسبت دهد.
- ۲- **ثباتهای آدرس:** حاوی آدرس داده و دستورالعمل ها می باشد.
- ۳- **ثبات کد وضعیت:** بیتهای هستند که به عنوان نتیجه عمل ها توسط سخت افزار مقدار گذاری می شود.

(بخشهایی از آن برای کاربر قابل رویت نیست)

ثبات های آدرس

- ۱- **ثبات شاخص (X + مقدار پایه = آدرس موثر)**
- ۲- **ثبات اشاره گر قطعه:** حافظه به قطعاتی تقسیم شده و یک ثبات برای نگهداری آدرس پایه (محل شروع) قطعه استفاده می شود. ممکن است چند ثبات آدرس پایه وجود داشته باشد.
- ۳- **ثبات اشاره گر پشته:** ثباتی خاص جهت اشاره به بالای پشته در حافظه اصلی.

ثبات های کنترل وضعیت

- این ثبات قابل رویت برای کاربر نیست.
- برای کنترل عمل پردازنده به کار می روند.
- **ثبات کلمه وضعیت (PSW):** حاوی اطلاعات وضعیت.
- علاوه بر کد وضعیت شامل اطلاعات ذیل می باشد
- بیت فعال و غیر فعال کردن وقفه.
 - بیت حالت کاربر / سرپرست.
- تخصیص چند صد یا هزار کلمه از ابتدای حافظه برای مقاصد کنترلی متداول است

چرخه دستورالعمل:

- ۱- چرخه واکشی: یک دستور را از حافظه می خواند
- ۲- چرخه دستورالعمل: اجرای دستورالعمل واکشی شده

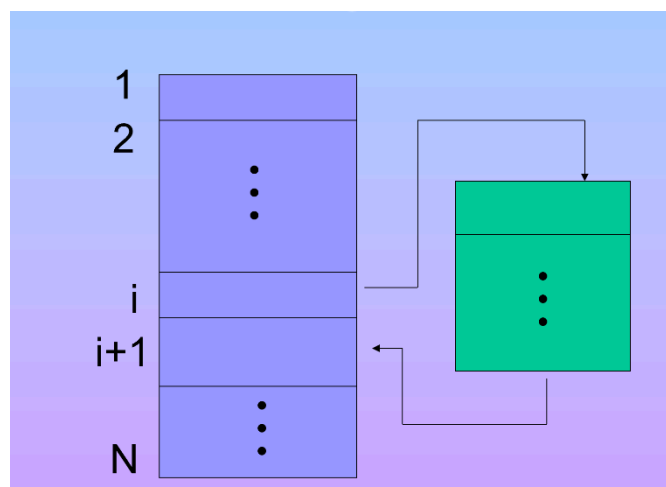
- ثبات pc : آدرس حافظه که با بیتی واکشی شده را نشان می دهد.
- واکشی دستورالعمل به ثبات IR
- معمولا پردازنده پس از واکشی دستورالعمل یک واحد به PC اضافه میکند.
- ثبات IR : دارای ۴ بیت که بیت اول عمل و سه بیت دیگر آدرس حافظه را مشخص می کند.
- **کد عمل** : عملی را که با بیتی پردازنده انجام بدهد را نشان می دهد.

وقفه: راهکاری است که به وسیله آن دستور عملی توسط پردازنده و پس از انجام عمل دیگر به روند عادی خود باز گردد.

رایجترین وقفه ها

- ۱- برنامه: وقفه ای است که در بعضی شرایط خاص یک دستورالعمل رخ می دهد مثل سر ریز، تقسیم بر صفر.
- ۲- زمان سنج: وقفه ای که توسط زمان سنج داخلی پردازنده تولید می شود.
- ۳- ورودی/خروجی: این وقفه به وسیله کنترل کننده ورودی و خروجی ایجاد می شود.
- ۴- نقص سخت افزار: این وقفه با سخت افزار تولید میشود.

وقفه



اعمال ریز پردازنده در یکی از چهار کرده زیر قرار دارد

۱- پردازنده - حافظه

۲- پردازنده - ورودی/خروجی

۳- پردازش داده ها (محاسباتی به منطقی)

۴- کنترل

پردازش یک وقفه

۱- یک دستگاه یک علامت وقفه برای پردازنده می دهد.

۲- اجرای دستورالعمل جاری به پایان می رسد.

۳- پردازنده اعلام وصول وقفه می کند.

۴- محتوای ثبات (PSW) و pc در بالای پشته کنترل قرار می دهد.

۵- پردازنده بر اساس وقفه مقدار pc جدید را قرار می دهد.

۶- باقیمانده اطلاعات وضعیت فرایند را ذخیره می کند.

۷- وقفه را پردازش می کند.

۸- بازبازی اطلاعات ثباتهایی که قبلا ذخیره شده.

۹- pc و psw قدیمی را دوباره پردازش کن.

وقفه های چند گانه:

(همزمانی چند وقفه)

دارای دو رویکرد:

۱- پردازش ردیفی وقفه: در هنگام وقوع یک وقفه بقیه وقفه ها از کار بیفتند تا کار آن تمام شود.

نکته منفی: اولویت نسبی به حساب نمی آید.

۲- پردازش وقفه تو در تو: در صورتی که وقفه دیگری رخ داد اولویت بالایی داشت به برنامه برگرداننده و

وقفه اول متوقف می شود.

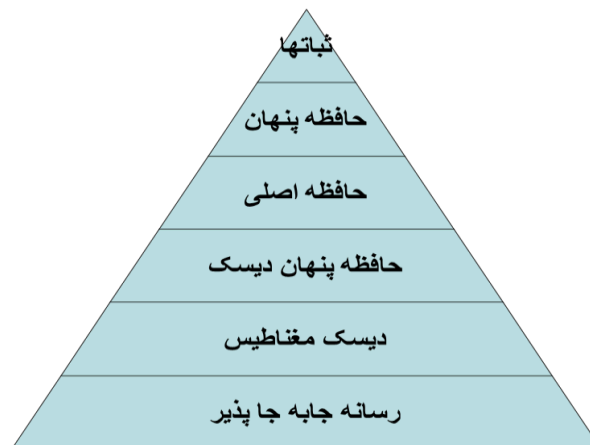
چند برنامه‌نگی:

چون پردازنده نسبت به قسمت های دیگر سریعتر است برای استفاده حداکثر از آن می توان در یک زمان چند برنامه از کاربر فعال باشد.

چند برنامه‌گی:

چند برنامه برای اجرا نوبت بگیرند.

سلسه مراتب حافظه



توضیحات درباره نمودار فوق

با حرکت به سطوح پایین تر این سلسله مراتب:

الف: کاهش هزینه در هر بیت

ب: افزایش ظرفیت

ج: افزایش زمان دسترسی

د: کاهش تعداد دفعات دسترسی پردازنده به حافظه

ثباتها: سریعترین و گرانترین و کوچکترین و ناپایدارند.

حافظه اصلی: دارای آدرس یکتا و ناپایدار هستند. با حافظه پنهان توسعه داده می شوند.

حافظه پنهان: قابل رویت برای کاربران نیست و ناپایدارند.

حافظه پنهان :

این حافظه به دو صورت **کارایی** را افزایش می دهند:

۱-نوشتن های روی دیسک دسته بندی می شوند.

۲-بعضی از داده ها قبل از نوشتن به وسیله برنامه هایی مورد مراجعه قرار می گیرند.

نقش حافظه پنهان

ارائه سریعترین حافظه موجود

حافظه ای بزرگ از انواع ارزان تر حافظه های نیمه هادی

حاوی بخشی از حافظه اصلی است.

اصول حافظه پنهان

نحوه عملکرد:

- ابتدا بررسی می شود که آیا کلمه در حافظه وجود دارد یا خیر؟
- اگر خیر بود یک بلوک را اختیار کرده و در آن قرار می دهیم.
- اگر بله بود به پردازش تحویل داده می شود.

طراحی حافظه پنهان :

نکات کلیدی:

- اندازه حافظه پنهان:** با وجود اندازه کوچک تاثیر زیاد.
- اندازه بلوک:** واحدی از داده ها که بین حافظه اصلی و پنهان مبادله می شود.
- تابع نگاشت:** هنگام فراخوانی بلوک جدید پدید می آید.
- الگوریتم تعویض:** بلوکی که باید عوض شود را انتخاب می کند (هنگام تعویض و باید دقت شود که کمترین استفاده در آینده را داشته باشد).
- سیاست نوشتن:** زمان انجام عمل نوشتن را مشخص می کند.

روشهای انتقال ورودی / خروجی

سه روش زیر وجود دارد:

- ۱- ورودی/ خروجی برنامه سازی شده.
- ۲- ورودی/ خروجی مبتنی بر وقفه.
- ۳- دسترسی مستقیم به حافظه (DBA)

ورودی/خروجی برنامه سازی شده:

با مواجهه با دستور العمل ورودی / خروجی صادر می شود.
عمل در خواست شده را انجام و بیت های مناسب از ثبات ذکر شده را مقدار گذاری می کند.
مسئول استخراج داده ها از حافظه به دستگاه خروجی .
مسئول ذخیره سازی داده ها در حافظه اصلی.

گروههای دستورالعمل های ورودی /خروجی :

- ۱- کنترل: دستورالعمل ها برای فعال کردن دستگاه خارجی.
- ۲- آزمون: بررسی مولفه های ورودی و خروجی.
- ۳- خواندن/نوشتن: دستور العمل هایی برای انتقال داده ها بین ثباتهای پردازنده و دستگاههای خارجی.

ورودی خروجی مبتنی بر مولفه:

- ۱- ابتدا یک READ به مولفه ورودی/خروجی می دهد.
 - ۲- سپس متن مورد نظر را ذخیره می کند.
 - ۳- پس از آماده شدن مولفه ورودی / خروجی به پردازنده وقفه می دهد.
 - ۴- پس متنی که فرمان READ را صادر کرده بار گذاری شده و اجرا می شود.
- دسترسی مستقیم به حافظه**
- ۱- دسترسی مستقیم به حافظه به وسیله مولفه دیگری روی گذرگاه سیستم انجام شود.
 - ۲- دسترسی مستقیم بر حافظه به عهده یک مولفه ورودی/خروجی گذاشته شود.
- حافظه $\longleftrightarrow$ DMA $\longleftrightarrow$ مولفه ورودی/خروجی

فصل دوم

نگاه کلی به سیستم عامل

وظایف سیستم عامل

- اجرای برنامه های کاربردی را کنترل می کند.
- رابط کاربر و سخت افزار.

سه هدف سیستم عامل

- ۱- سهولت: استفاده از کامپیوتر ساده شود
- ۲- کارآمدی: استفاده کارآمد از منابع
- ۳- قابلیت رشد: سبب توسعه و آزمایش سیستم می شود

برنامه های سودمند

- مجموعه برنامه های سیستمی هستند.
- توابعی هستند که به دفعات استفاده می شوند و به ایجاد برنامه مدیریت پروژه و کنترل دستگاه های ورودی و خروجی کمک می کنند.
- سیستم عامل مهمترین برنامه سیستمی است.

وظایف سیستم عامل

- ایجاد برنامه
- دسترسی به دستگاه های ورودی و خروجی
- کنترل دسترسی به پرونده ها
- دسترسی به سیستم عامل
- کشف و پاسخ به خطا
- حسابداری

سیستم عامل به عنوان مدیر منابع

- سیستم عامل مسئول مدیریت انتقال و ذخیره سازی و پردازش داده ها را دارد.
- سیستم عامل به عنوان راهکار کنترلی ازدو جهت غیر عادی است:
- ۱- سیستم عامل مانند نرم افزار عادی کار می کند.
 - ۲- سیستم عامل مرتبا کنترل کردن را رها می کند.

هسته سیستم عامل :

بخشی که در حافظه اصلی است.
بیشترین دفعات استفاده را دارد.
تخصیص حافظه مشترکا توسط سیستم عامل و سخت افزار مدیریت حافظه در پردازنده کنترل می شود.

در مورد زمان استفاده از I/O تصمیم می گیرد.
میزان تخصیص پردازنده به یک کار را مشخص می کند.

دلایل تغییر سیستم عامل

- ۱- ارتقاء و انواع جدید سخت افزار .
- این امکانات نیازمند حمایت پیچیده تری از طرف سیستم عامل است.
- ۲- خدمات جدید: برای پاسخگویی به نیازهای کاربران.
- ۳- رفع خطا: ساخت تیم مولفه ای است و رابط ها باید مشخص شود.

تکامل تدریجی سیستم عامل

- ۱- پردازش ردیفی.
- ۲- سیستم های دسته ای ساده.
- ۳- سیستم های چند برنامه ای دسته ای.
- ۴- سیستم های اشتراک زمانی.

پردازش ردیفی

- به علت دسترسی به کامپیوتر به صورت ردیفی این نام را دارند.
- در سال ۱۹۴۰-۱۹۵۰ به وجود آمد.
- کاربر با سخت افزار در ارتباط بود.
- سیستم عامل وجود نداشت.
- دارای یک میز فرمان بود.

- به زبان ماشین و به وسیله دستگاه ورودی با گذاری می شد.
- خروجی در چاپگر ظاهر می شد.

مسئله اصلی سیستم های اولیه

۱-زمانبندی :

توسط برگهای نوبت گیر برای ماشین وقت می گرفتند و در زمان پیش بینی شده کارش تمام نمی شد.

۲-زمان نصب :

اگر در زمان کار برای هر یک از کار ها مشکلی ایجاد شود باید از اول کارش را شروع کند.

سیستم های دسته ای:

- اولین سیستم عامل دسته ای (۱۹۵۰) به وسیله General motors و برای استفاده IBM 701 به وجود آمد.
- پردازش دسته ای ساده توسط نرم افزار ناظر.
- کاربر دسترسی مستقیم به ماشین ندارد.
- کاربر کار را توسط نوار به متصدی کامپیوتر می دهد.
- متصدی همه کارها را روی دستگاه ورودی گذاشته و وارد کامپیوتر می شود.
- ناظر به طور خودکار بار کردن برنامه بعدی را به عهده دارد.

ناظر

- قسمت اعظم ناظر: در حافظه اصلی و آماده اجرا می باشد

(ناظر ماندگار)

- بقیه ناظر: برنامه سودمند و توابع عمومی و مشترک

(زیر برنامه)

زمان تنظیم شرایط اولیه کارها را ناظر انجام میدهد.

- JCL: دستورالعمل هایی از زبان کنترل کار.

این دستورالعمل ها با \$ مشخص می شود.

نحوه عملکرد سیستم عامل دسته ای

خواندن از حافظه جانبی برنامه کاربر در حافظه جانبی
ناظر ← ← کارت FTN \$ بار کردن مترجم ← کدمقصد ← ذخیره

بوسیله ناظر احضار
کارت \$LOAD ← خوانده ← بارکننده ← برنامه مقصد ← مترجم (در حافظه اصلی)
← کنترل به مترجم مقصد.

مزایای سیستم عامل دسته ای

- ۱- حفاظت حافظه اصلی.
- ۲- زمان سنج:
- جلوگیری از انحصار سیستم توسط یک کاربر.
- ۳- دستورالعمل های متمایز:
- دستورالعمل هایی که توسط ناظر صادر می شود.
- ۴- وقفه ها:
- این خصوصیت به سیستم انعطاف می دهد.

سیستم های چند برنامه ساده

- حافظه را برای سه یا چهار برنامه تخصیص داده.
- موضوع اصلی سیستم عامل های امروزی است.
- باعث افزایش سرعت کار می شود
- به خصوصیت سخت افزار تکیه دارد.
- از سیستمهای تک برنامه ای بیجیده تر است
- نیازمند نوعی مدیریت حافظه است

سیستم های اشتراک زمانی

- از چند برنامه‌گی برای رسیدن به حالت محاوره ای استفاده می کند.
- پردازنده بین کاربران به اشتراک گذاشته می شود.
- داشتن کاربران متعددی که از طریق پایانه خود به طور هم زمان از سیستم عامل استفاده می کنند.
- اگر N کاربر داشته باشد هر کاربر $1/N$ از زمان مفید پردازنده استفاده می کنند.

چند برنامه‌گی دسته ای در مقابل اشتراک زمانی

اشتراک زمانی	چند برنامه‌گی دسته ای	
حداقل زمان پاسخ	حداکثر استفاده از پردازنده	هدف اصلی
فرمان هایی که از پایانه وارد می شود	دستورالعملهای زبان کنترل کار که همراه کار ارائه شده است	منبع دستورات به سیستم عامل

دستاوردهای اصلی

پنج دستاورد توسعه سیستم عامل:

- ۱- فرآیندها.
- ۲- مدیریت حافظه.
- ۳- حفاظت اطلاعات و ایمنی.
- ۴- زمانبندی و مدیریت منابع.
- ۵- ساختار سیستم.

فرآیندها

- یک مفهوم بنیادی در سیستم عامل است.
- یک برنامه در حال اجرا است.
- روح متحرک یک برنامه است.

- موجودیتی که می توان به یک پردازنده داده شود و روی پردازنده اجرا شود.

سه محور برای ایجاد و توسعه سیستم کامپیوتر:

۱- چند برنامه‌گی:

برای مشغول نگهداشتن هم زمان پردازنده و خروجی.

۲- اشتراک زمانی:

آماده پاسخگویی هم زمان به چند کاربر.

۳- سیستم های تراکنش بلا درنگ:

تعدادی از کاربران در حال وارد کردن درخواستها یا تغییرات خود روی بانک اطلاعاتی باشند.

علت بروز خطاها در سیستم نرم افزاری:

۱- همگام سازی نا مناسب:

به سبب بروز یک حادثه یک کار عقب بیفتد.

۲- شکست در انحصار متقابل:

استفاده بیش از دو کاربر از یک منبع به طور هم زمان.

۳- عملکرد غیر قطعی برنامه:

ممکن است کاربرها در کار یک دیگر دخالت کنند.

۴- بن بستها:

دو یا چند برنامه به خاطر هم معلق بمانند.

اجزاء یک فرآیند:

۱- یک برنامه قابل اجرا.

۲- داده های مورد نیاز این برنامه.

۳- متن یا وضعیت اجرای آن برنامه.

(اصلی است و شامل تمام اطلاعاتی است که یک سیستم عامل برای مدیریت فرآیندها احتیاج دارد).

- متن: شامل محتویات ثباتها و...

- برای هر فرآیند یک مدخل برای فهرست فرایندها وجود دارد.

- ثبات شاخص: شامل فهرستی از فرآیندها که پردازنده را استفاده می کنند.

- ثبات شمارنده برنامه: برنامه بعدی که قرار است اجرا شود.

ثبات های پایه و حد

- در ناحیه ای از حافظه قرار دارند.
- شمارنده برنامه و تمام مراجعات به داده هانسیبت به محتویات ثبات پایه تفسیر می شود و نباید از ثبات حد بیشتر باشد.
- از دخالت بین فرایندها جلوگیری می کنند.

پنج مسئولیت اصلی در مدیریت حافظه:

- ۱- جداسازی فرآیندها: از دخالت فرآیندها جلوگیری می کند.
- ۲- تخصیص و مدیریت خودکار: باید به مدیریت ها جا دهد.
- ۳- حمایت از برنامه سازی مولفه ای: باید مولفه های برنامه را تعریف کند.
- ۴- حفاظت و کنترل دسترسی: باید به کاربر اجازه دهد تا به بخشهایی از حافظه دسترسی داشته باشند.
- ۵- حافظه دراز مدت: ذخیره اطلاعات برای مدت طولانی

حفاظت از اطلاعات و ایمنی:

- به سه گروه زیر تقسیم می شود:
- ۱- کنترل دسترسی: تنظیم دسترسی کار به کل سیستم.
 - ۲- کنترل جریان اطلاعات: تنظیم جریان اطلاعات در داخل سیستم و تحویل به کاربر.
 - ۳- گواهی: به صحت اطلاعات.

زمانبندی و مدیریت اطلاعات:

- سه عامل زیر را در نظر می گیرند:
- ۱- انصاف: قرار دادن منابع به هر یک از پردازنده ها به طور عادلانه.
 - ۲- حساسیت در مقابل تفاوت ها: باید به خواسته ها جوابگو باشد.
 - ۳- کارایی: باید بکوشد تا توان عملیاتی را حداکثر زمان پاسخ را حداقل و در مورد سیستمهای اشتراکی زمانی از حداکثر تعداد کاربران حمایت کند.

ساختار سیستم:

اندازه یک سیستم سه مولفه نا مطلوب را در بر دارد:

- ۱- سیستم عامل از نظر زمان تحویل تاخیر دارد.
- ۲- سیستم ها اشکالهای پنهانی دارند که در هنگام کار اشکال خود را نشان می دهند.
- ۳- کارایی آن از حد انتظار کم تر است.

برای رفع مشکل:

- سیستم های کوچک ساختار مولفه ای.
- سیستم های بزرگ ساختار سلسله مراتبی.
- ساختار سلسله مراتبی عملکرد سیستم را بر اساس پیچیدگی و ویژگی زمانی و سطح تجزید آنها تفکیک می کند.
- دارای چند سطح است که سطوح به هم وابستگی دارند که اعمال ابتدایی را انجام و جزئیات را پنهان می کند.
- سطوح پایین با اندازه کوتاه تر زمان سر کار دارد.
- بعضی مستقیما با سخت افزار در تبادل اطلاعات هستند.

سطوح سیستم عامل سلسله مراتبی:

- سطح ۱: شامل مدارات الکتریکی.
- سطح ۲: مجموعه دستورالعمل پردازنده.
- سطح ۳: مفهوم رویه یا زیر برنامه.
- سطح ۴: وقفه ها.
- سطح ۵: فرآیند (یک برنامه در حال اجرا)
- سطح ۶: حافظه ثانویه کامپیوتر.
- سطح ۷: یک فضای آدرس منطقی برای فرآیند.
- سطح ۸: مبادله اطلاعات و پیامها بین فرآیند.
- سطح ۹: ذخیره سازی دراز مدت پرونده ها.
- سطح ۱۰: دسترسی به دستگاه های خارجی.
- سطح ۱۱: نگهداری رابط بین شناسه های خارجی و داخلی.
- سطح ۱۲: حمایت اطلاعات لازم برای مدیریت فرآیند ها.

- سطح ۱۳: واسط کاربر و سیستم عامل. (پوسته)

ویژگی های سیستم عامل های جدید:

- معماری ریز هسته.
- چند نخی.
- چند پردازشی متقارن.
- سیستم های عامل توزیعی.
- طراحی شیء گرا.

معماری ریز هسته

موارد واگذاری به هسته:

- ۱- دسترسی به فضای آدرس.
- ۲- ارتباط بین فرآیندها.
- ۳- زمانبندی پایه ای.

رویکرد ریز هسته:

- ۱- پیاده سازی را ساده می کند.
- ۲- موجب انعطاف می گردد.
- ۳- محیط توزیعی کاملاً سازگار.

چند نخی

- فرآیندها به نخ هایی تقسیم می شوند که می توانند به طور هم زمان اجرا شوند
- در فرآیندهایی که چند وظیفه اصلی مستقل را انجام می دهند مفید است.
- فرایند: مجموعه یک یا چند نخ و منابع سیستمی تخصیص داده شده (مشابه یک برنامه حال اجرا)

چند پردازشی متقارن

- ۱- پردازنده متعددی وجود دارد.
- ۲- از امکانات ورودی / خروجی و حافظه اصلی به طور مشترک استفاده می کنند.
- ۳- تمام پردازنده ها اعمال یکسانی را می توانند انجام دهند.

امتیازات چند پردازی متقارن نسبت به معماری تک پردازنده:

- کارایی:
قرار دادن ترتیب کارها به نحوی که به موازات هم انجام شوند.
- دسترسی پذیری:
از آنجا که تمام پردازنده ها یک عمل را انجام می دهد خرابی در یکی از آنها موجب توقف ماشین نمی شود.
- رشد:
با اضافه کردن پردازنده.
- مقیاس پذیری:
قیمت ها متناسب با تعداد پردازنده ها باشد.

سیستم های عامل توزیعی

تصور یک فضای حافظه اصلی واحد و یک فضای حافظه ثانوی واحد و دیگر امکاناتی که دسترسی به آنها یکنواخت شده است را ارائه می کند.

طراحی شیء گرا

افزودن ملحقات مولفه ای به یک هسته کوچک.
محیا ساختن ابزار های توزیعی و سیستم عامل های توزیعی را آماده میکند.

نگاهی کلی به WINDOWS NT

- سیستم عامل چند کاربره چند وظیفه ای.
- برای کامپیوتر شخصی طراحی شده.
- اساسا از صفر طراحی شده است.
- این سیستم عامل منسوب به MS-DOS یا PC XT است که توسط شرکت MICROSOFT برای اولین بار کامپیوتر IBM را ایجاد کرد.
- این سیستم عامل از دیسک سخت حمایت می کرد.
- با بهره گیری از توان ریز پردازنده های جدید محیط تک کاربره و چند وظیفه ای کاملی به وجود آورد.

- اولین گونه در سال ۱۹۹۳ وبا همان واسط گرافیکی WINDOWS3.1 ارائه گردید.
- سپس گونه NT3.X و NT4.0 را ارائه کرد.
- (تغییر در مولفه های گرافیکی)
- و در نهایت NT 5.0 را بیرون دادند.
- (دارای ویژگی افزایش خدمات برای توزیعی است)

معماری NT

- NT نرم افزار کاربردی را از نرم افزار سیستم عامل جرا می کند.
- نرم افزار سیستم عامل که به ان مجری NT می گویند در حالت هسته اجرا می شوند.
- به داده های سیستم و به سخت افزار سیستم دسترسی دارد.
- نرم افزار باقیمانده که در حالت کاربر اجرا می شود به داده های سیستم دسترسی محدود دارد.

مجری NT:

- یک معماری ریز هسته تغییر یافته است.
- دسترسی از طریق مولفه مسئول امکان دارد.
- دسترسی به داده های کلیدی از طریق توابع مناسب.
- می توان مولفه ها را تغییر داد یا باز نویسی مجدد سیستم و...
- قابلیت حمل دارد.
- دید یکسان به سخت افزار مورد استفاده دارد.

مولفه های خدمات سیستم:

- مدیر ورودی و خروجی:
- در خواستها را به ترتیب اولویت رسیدگی می کند.
- مدیر شیء:
- قواعدی را برای نگهداری و نام گذاری و.. شیءها اعمال می کند.
- ناظر ایمنی:
- مسئول اعمال قواعد اعتبار سنجی.
- مدیر فرآیند:
- ایجاد و حذف شیء ها.

- امکان فراخوانی رویه ورودی:
- فراخوانی رویه از راه دور.
- مدیر حافظه مجازی:
- نگاشت بین آدرس های مجازی.
- مولفه های پنجره ها / گرافیک:
- ایجاد واسط پنجره در صفحه نمایش.

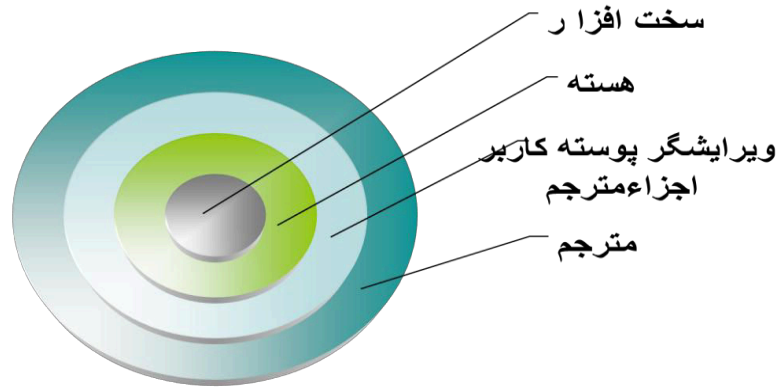
مهمترین مفاهیم شیء گرا که در NT به کاررفته:

- در برگیری: هرشیء دارای یک یا چند فقره داده است که خصیصه نام دارد
- کلاس شیء و رویداد: الگویی که خدمات را فهرست می کند.
- وراثت: در داخل مجری حمایت شده است.
- چند شکلی: مجموعه مشترکی از توابع API برای پردازش و...

سیستم UNIX

- یک سیستم چند کاربره
- برای کامپیوتر های کوچک به وجود آمد
- سخت افزار توسط نرم افزار کامپیوتر احاطه شده است
- این بخش را هسته گویند
- برای اجرای روی یک پردازنده واحد طراحی شده.
- قادر به حفاظت از ساختمان داده در مقابل دسترسی هم زمان چند پردازنده نیست
- هسته آن چند تطبیق پذیر نیست
- برای توسعه پذیری طراحی نشده
- امکان کمی برای استفاده مجدد از کد دارد.

بخشهای سیستم UNIX



فصل سوم

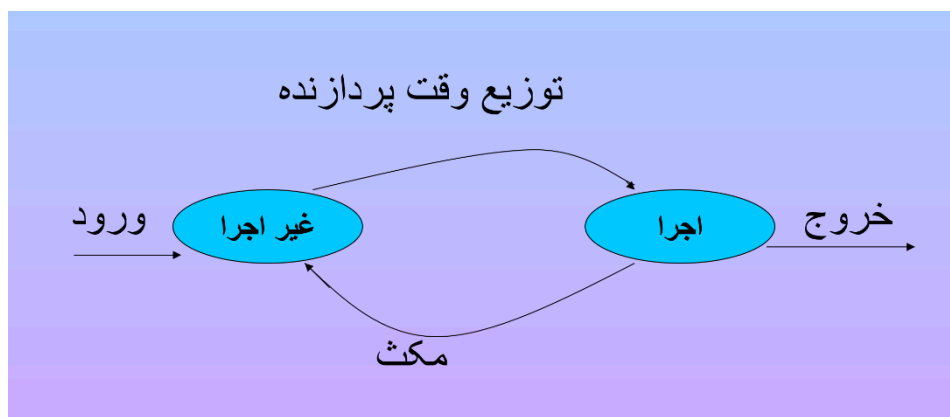
شرح و کنترل فرآیند

- تمام نیازمندی های اصلی که سیستم عامل باید پاسخگو باشد به وسیله فرایند قابل بیان است.
- سیستم عامل باید در بین اجرای فرآیند قرار گیرد.
- سیستم عامل باید با بهره گیری از یک سیاست خاص منابع در اختیار سیستم قراردادده واز وقوع بن بست جلوگیری کند.
- سیستم عامل باید از ارتباط بین فرایند هاو ایجاد فزایندها توسط کاربر حمایت کند.

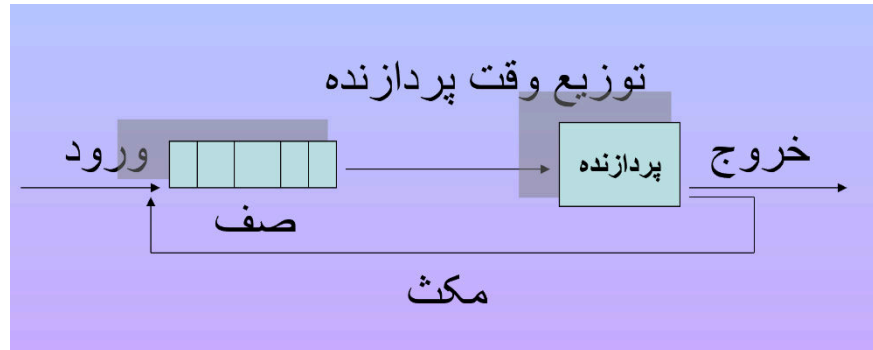
حالات فرآیند:

- اساسی ترین عمل پردازنده اجرای دستورالعمل های موجود در حافظه است.
- اجرا شامل دنباله ای از دستورالعمل های همان برنامه است.
- به اجرای یک فرایند خاص فرایند یا وظیفه گویند.
- رفتار یک فرآیند به خصوص را می توان با فهرست کردن دنباله دستورالعمل ها یی که برای آن فرآیند اجرا می شود مشخص نمود که به آن رد آن فرآیند گویند.
- مسئولیت اصلی سیستم عامل کنترل یک فرآیند است.
- یک فرآیند ممکن است در یکی از دو حالت اجرا و غیر اجرا باشد.
- وقتی یک سیستم عامل فرآیندی را دریافت می کند آن را در حالت غیر اجرا قرار می دهد.
- پس از رسیدن نوبت فرآیند فعلی را به حالت غیر اجرا برده و فرآیند را به حالت اجرا برده.
- در مورد هر فرآیندی باید اطلاعاتی را ذخیره کرد.
- فرآیندی که در حال اجرا نیست باید در صفی به انتظار نوبت قرار گیرد.
- گاهی صفی است که حاوی جداولی است که وضعیت فرآیند ها را نمایش می دهد.
- وقتی یک فرآیند در معرض وقفه قرار می گیرد به صف انتظار می رود.
- اگر یک فرآیند کارش تمام شود کنار گذاشته می شود.

نمودار تغییر حالت:



نمودار صف بندی



ایجاد و پایان فرآیند:

- عمر یک فرآیند محدود به زمان ایجاد تا زمان خاتمه دادن به آن است.
- در هنگام ایجاد سیستم عامل ساختمان داده مربوط به آن فرآیند را ساخته و فضای آدرس مربوط به آن را تخصیص می دهد.

معمولا چهار حادثه منجر به ایجاد فرآیند می گردد:

- ۱- در محیط دسته ای: یک فرآیند جدید در پاسخ به یک کار
- ۲- برقراری ارتباط محاورهای: کار از طریق پایانه با سیستم ارتباط برقرار کند
- ۳- ارائه یک خدمت به وسیله سیستم عامل: فرآیندی را برای خدمت ایجاد کند
- ۴- زایش توسط فرآیند موجود: برای بهره گیری از توازی یا تفکیک.

پایان فرآیند:

هر کار دسته ای باید حاوی دستورالعمل توقف یا فراخوانی صریح یک خدمت سیستمی برای پایان باشد.

در بعضی از سیستم عامل ها ممکن است فرآیندی به وسیله فرآیند ایجاد کننده اش و با پایان فرآیند پدرش نیز پایان یابد.

- زایش فرزند:

فرآیندی با درخواست فرآیند دیگر به وجود می آید

- فرزند:

فرآیند زاینده را پدر و فرآیند ایجاد شده را فرزند گویند

انواع خطا:

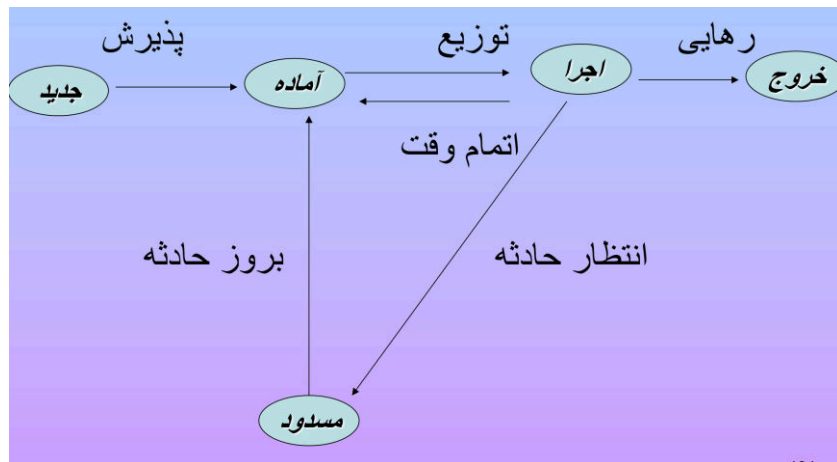
- خطای محاسباتی: محاسبات غیر مجاز
- گذشت زمان: بیش از حد برای بروز حادثه منتظر مانده
- خطای ورودی و خروجی

و...

پنج حالت ممکن برای فرآیند:

- اجرا: فرآیندی هم اکنون در حالت اجرا است.
- آماده: با گرفتن فرصت به اجرا درمی آیند.
- مسدود: با تمام شدن وقت و یا بروز حادثه اتفاق می افتد.
- جدید: فرآیندی که هم اکنون ایجاد شده.
- خروج: به علت دستور توقف یا به دلیلی قطع شده.

مدل پنج حالتی برای فرآیند:



انواع حوادثی که منجر به تغییر حالت شده:

- تهی ← آماده: فرایند جدیدی را برای اجرای یک برنامه ایجاد می کند.
- جدید ← آماده: آماده: آمادگی برای گرفتن یک برنامه.
- آماده ← اجرا: زمان انتخاب یک فرآیند رسیده.
- اجرا ← خروج: فرآیند جاری اعلام می کند که تمام شده.
- اجرا ← آماده: اتمام زمان مجاز برای یک فرآیند .

(متداولترین)

- اجرا ← مسدود: درخواست یک منبع با انتظار.
- مسدود ← آماده: حادثه مورد نظر اتفاق افتاده.
- آماده ← خروج: با تصمیم پدر.
- مسدود ← خروج: توضیح تغییر حالت قبل در اینجا نیز صادق است.

در اکثر اوقات پردازنده بی کار می ماند برای این کار راه حل های زیر پیشنهاد می شود:

۱- حافظه اصلی را گسترش داد: که دو عیب دارد:

۱- با افزایش هزینه همراه است.

۲- درخواست برای حافظه زیاد شده.

۲- مبادله: انتقال بخشی از یک فرآیند به دیسک.

هنگامی که هیچ یک از فرآیندها در صف آماده نیستند منع لازم را دریافت نکرده به حالت معلق می روند. و فرآیند دیگری از حالت معلق به حالت آماده برده و اجرا می شود.

عیب:

مبادله یک ورودی و خروجی است.

امتیاز:

ورودی و خروجی عموماً سریعتر از سیستمی است و باعث افزایش کارایی می شود.

فرآیند معلق:

- فوراً آماده اجرا نیست.
- یک فرآیند ممکن است منتظر یک حادثه باشد یا نباشد.
- توسط عاملی در حالت معلق گذاشته شده تا از اجرای آن جلوگیری کند.
- منتظر فرمان باشد.

دلایل تعلیق فرآیند:

- مبادله: سیستم عامل نیاز به حافظه کافی دارد.
- دلایل دیگر سیستم عامل: فرآیندی را که سودمند یا لازم و... است را معلق کند.
- درخواست کاربر محاورهای: کاربر به منظور استفاده از منبع برنامه را معلق کند.
- ترتیب زمانی: فرآیند به طور دوره ای اجرا شود.
- درخواست فرآیند پدر: فرآیند فرزند را به تعلیق بیندازد.

ساختارهای کنترلی سیستم عامل:

برای کنترل فرآیندها باید اطلاعات مربوط به فرآیند در جدولی ذخیره شود. تمام سیستم عامل ها اطلاعات در چهار گروه ذخیره می کنند.

از جداول حافظه برای دنبال کردن اطلاعات حافظه اصلی و ثانویه استفاده می کنند ، قسمتی برای سیستم عامل و یقیه برای فرآیند

جداول حافظه باید حاوی اطلاعات زیر باشد:

- ۱- تخصیص حافظه اصلی به فرآیند.
- ۲- تخصیص حافظه ثانویه به فرآیند.
- ۳- ویژگی های حفاظتی فرآیند.
- ۴- اطلاعات مورد نیاز برای مدیریت حافظه مجازی.

انواع جداول:

- **جداول ورودی / خروجی:** برای نگهداری و مدیریت دستگاه های ورودی و خروجی و... استفاده می شود.
- **جداول پرونده:** اطلاعات مربوط به پرونده های موجود، محل آنها در حافظه ثانویه ،وضعیت جاری و...نگهداری کنند.
- **جداول فرآیند:** مدیریت فرآیندها استفاده می شود.

ساختار کنترلی فرآیند:

هر فرآیند دارای یک صفات است که معمولا همراه آن به سیستم عامل می آید که به مجموعه این صفات بلوک کنترل فرآیند می گویند.

به مجموعه داده برنامه، داده ها ،پشته و صفت ،تصویر فرآیند می گویندو محل آن به مدیر حافظه بستگی داردو در حافظه ثانویه نگهداری می شود و برای اجرا باید به حافظه اصلی برود.

اجزای متداول تصویر یک فرآیند:

- داده های کاربر: بخش قابل تغییر فضای کاربر.
- برنامه کاربر: برنامه ای که قرار است اجرا شود.
- پشته سیستم: برای ذخیره پارامترها و آدرس و...
- بلوک کنترل فرآیند: اطلاعات لازم برای کنترل فرآیند.

اطلاعات مربوط به یک بلوک فرآیند حاوی:

۱- شناسایی فرآیند

۲-اطلاعات وضعیت پردازنده

۳-اطلاعات کنترل فرآیند

۱-شناسایی فرآیند: یک شناسه عددی یکتا نسبت می دهند.

مزایای شناسه:

مراجعات به جداول تحت کنترل است.

از طریق شناسه می توان پی به برقراری ارتباط بین فرآیندها برد.

به فرآیندی یک شناسه کاربر نسبت داد.

۲-اطلاعات وضعیت پردازنده:

محتوی ثباتهای پردازنده می باشد.

(PSW) برای اطلاعات وضعیت می باشد.

حاوی کد شرایط و دیگر اطلاعات است.

۳-اطلاعات کنترل فرآیند:

بلوک کنترل فرآیند گفته می شود.

موجب ایجاد هماهنگی بین فرآیندهای فعال می شود.

نقش بلوک کنترل فرآیند:

- بلوک کنترل فرآیند نقش مهم در ساختمان سیستم عامل دارد.
- وضعیت سیستم را تعریف می کند.
- از شناسه یکتای فرآیند به عنوان شاخص استفاده می کنند.

مشکلات بلوک کنترل فرایند برای حفاظت

- ۱- وجود اشکال در یک بلاک منجر به لطمه زدن به تمام بلاک ها می شود.
- ۲- تغییر در یک بلوک منجر به تغییر در بسیاری از مولفه ها می شود.

کنترل فرآیند:

• حالت اجرا:

- به سیستم عامل مربوط می شود.
 - بعضی از دستورات تنها در حالت ممتازتر قابل اجرا هستند.
 - به حالت کم امتیازتر حالت کاربر می گویند.
- به حالت متمایز، حالت سیستم، حالت کنترل یا حالت هسته گویند که نرم افزار کنترل پردازنده ؛ دستورالعملها؛ ثباتها ؛ حافظه را دارد.
- یک بیت برای حالت اجرا وجود دارد که بیت حالت به بیت هسته واگذار می شود PSW در ثبات و دستورالعمل حالت را تغییر می دهد.

ایجاد فرآیند:

- تخصیص شناسه یکتا به فرآیند.
- تخصیص فضا برای تمام اجزای تصویر فرآیند.
- بلوک کنترل فرآیند مقدار گذاری می شود.
- برقراری پیوندهای لازم.
- ایجاد و گسترش ساختمان داده های دیگر.

تعویض فرآیند:

ابتدا باید وقفه های سیستم را در نظر بگیریم که به نوع دیگر تله می باشد.

وقفه نوع اول : مستقل از فرآیند در حال اجرا حاصل می شود.

وقفه نوع دوم : به خطا یا شرایط استثنایی مربوط است.

با هر وقفه معمولی، ابتدا کنترل به یک گرداننده وقفه منتقل می شود و به یک روال که مخصوص سیستم عامل آن وقفه است منتقل می شود.

وقفه ساعت: فرایند جاری تعویض و فرآیند دیگر بار گذاری

وقفه ورودی / خروجی: دادن ورودی / خروجی به یک فرایند.

نقص حافظه: اشتباه بودن آدرس.

تغییر حالت فرآیند:

- ذخیره سازی متن پردازنده.
- بهنگام سازی بلوک های کنترل فرایندی در حالت اجرا.
- انتقال بلوک فرآیند مربوط به این فرآیند به صف مناسب.
- انتخاب فرآیند دیگر برای اجرا.
- بهنگام سازی بلوک های کنترل فرایند انتخاب شده.
- بهنگام سازی ساختمان داده های مدیریت حافظه .
- بار گذاری مجدد متن پردازنده.

مدیریت فرآیند در UNIX SVR4:

بخش اعظم سیستم عامل، در داخل محیط یک فرآیند کاربردی اجرا می شود.

از دو گروه فرآیند سیستمی و کاربردی استفاده می کند.

فرآیند های سیستمی در کد هسته اجرا می شود.

فرآیند های کاربر در حالت کاربر برای اجرای برنامه های کاربر.

حالات فرآیند در UNIX:

- اجرای کاربر
- اجرای هسته
- آماده در حافظه (بعد از زمانبندی حافظه)
- خفته و در حافظه (تا بروز حادثه قابل اجرا نیست)
- آماده اجرا و مبادله شد (به حافظه اصلی منتقل می شود)
- قبضه شده (از هسته به کاربر می رود)
- ایجاد شده (برای اجرا آماده نیست)
- جادویی (فرآیندی دیگر وجود ندارد)

تصویر فرآیند در UNIX:

- هر فرآیند مجموعه ای از ساختمان داده هایی است که تمام اطلاعات لازم برای مدیریت و توزیع وقت پردازنده و فرآیند را در اختیار سیستم عامل قرار میدهد.
- جدولی که در صفحه بعد وجود دارد، عناصر تصویر فرآیند را که سه بخش متن سطح کاربر، متن ثابت و متن سطح سیستم سازماندهی شده، خلاصه کرده است.
- **متن سطح کاربر:** حاوی عناصر پایه ای کاربر است.
- **متن ثابت:** هنگامی که فرآیندی در حال اجرا نیست در آن ذخیره می شود.
- **متن سطح سیستم:** حاوی اطلاعاتی برای مدیریت سیستم است و شامل یک بخش پویا و یک بخش ایستا است.

متن سطح کاربر دستورات قابل اجرا داده های قابل دستیابی حاوی نشانوند و ... کاربر حافظه مورد اشتراک با دیگر	متن فرایند داده های فرآیند پشته کاربر حافظه مشترک
متن ثابت آدرس دستورالعمل کار بعدی حاوی وضعیت سخت افزار در زمان قبضه به بالای پشته کاربر یا هسته اشاره دارد وابسته به سخت افزار	شمارنده برنامه ثبات وضعیت پردازنده اشاره گر پشته ثباتهای همه منظوره
متن سطح سیستم وضعیت فرایند را تعریف می کند اطلاعات کنترل فرایند نگاشت از حافظه مجازی به آدرس فیزیکی حاوی قاب پشته	مدخل جدول فرآیند ناحیه کاربر جدول منطق هر فرآیند پشته هسته

مدخل جدول فرآیند در UNIX:

- اطلاعات کنترل فرآیند که همواره در دسترس هسته است را در بردارد.
- در حافظه اصلی قرار دارد.
- ناحیه کاربر (یا ناحیه U) حاوی اطلاعات اضافی کنترل است.
- این اطلاعات موقع صفحه بندی در حافظه اصلی قرار می گیرد.
- هسته UNIX همیشه در متن فرآیندی اجرا می گردد.
- جدول بعدی نمایشگر این اطلاعات خواهد بود.

وضعیت فرآیند اشاره گر ها اندازه فرآیند شناسه های کاربر شناسه های فرآیند توصیفگر حادثه اولویت علامت زمان سنج ها پیوند P وضعیت حافظه	حالت فعلی به ناحیه U و ناحیه فرآیند فضا را به سیستم عامل نشان می دهد کاربر مسئول را مشخص می کند شناسه پدر مقدار گذاری می شود به حات آماده می رود (در حات خفته) زمانبندی فرآیند شمارش علامتهای رسیدگی نشده زمان اجرای فرآیند و استفاده منبع و... اشاره به عنصر بعدی تصویر یا انتقال فذآیند به حافظه اصلی
---	--

ناحیه U در UNIX:

اشاره گر جدول فرآیند شناسه های کاربر زمان سنج ها آرایه گرداننده فرآیند پایانه کنترل فیلد خطا مقدار بازگشت پارامتر های I/O پارامترهای پرونده جدول توصیفگر پرونده کاربر فیلد های حد فیلد های حالت مجاز	مبین مدخلی که به ناحیه U مربوط است شناسه واقعی کاربر زمان لازم برای اجرا عکس اتعمل های فرآیند را مشخص می کند مبین پایه برقراری ارتباط پایانه با کامپیوتر برای کاربر خطا ها زا ثبت می کند حاوی نتیجه فراخوانی مقدار داده که باید منتقل شود و... جدول راهنمای جاری و... پرونده های باز شده را ثبت می کند
--	--

اندازه پرونده و فرآیند را که می توانید بنویسید را محدود می کند حالت پرونده های ایجاد شده را تنظیم می کند	
--	--

کنترل فرآیند:

- ایجاد فرآیند توسط (FORK) که یک فراخوان هسته سیستم است انجام می شود و اعمال زیر انجام می شود:
 ۱. برای فرآیند جدید، عنصری در جدول قرار می دهد.
 ۲. یک شناسه یکتا به فرزند می دهد.
 ۳. یک کپی از پدر ایجاد می کند.
 ۴. شماره پرونده های پدر را افزایش می دهد.
 ۵. فرآیند فرزند را در حالت اجرا قرار می دهد.
 ۶. شماره شناسه فرزند را پدر و مقدار صفر می دهد.
 - تمام این اعمال در حالت هسته فرآیند پدر انجام می شود.
- و موقع توزیع فرآیند یکی از اعمال زیر انجام می شود:
- ۱- ماندن در فرآیند پدر.
 - ۲- انتقال کنترل به فرآیند فرزند.
 - ۳- انتقال کنترل به فرآیند دیگر.

فصل چهارم

نخها، چندپردازی متقارن و ریز هسته ها

فرآیندها و نخ ها

فرآیند دو خصوصیت زیر را دارد:

- تملک منبع: (فرآیند یا وظیفه)

به هر فرآیند یک آدرس مجازی برای نگهداری تصویر فرآیند می دهد.

- توزیع وقت فرآیند: (نخ یا فرآیند سبک وزن)

فرآیند مسیر اجرای یک یا چند برنامه است و توسط سیستم عامل مورد زمان بندی و توزیع قرار می گیرد.

چند نخی:

- به قابلیت از سیستم که امکان اجرای چند نخ را حمایت می کند.
- تک نخ: به رویکرد سنتی یک نخ اجرا در هر فرآیند .
- مانند MS-DOS از فرآیند تک کاربره و از یک نخ حمایت می کند.
- UNIX از چند کاربر ولی تنها از یک نخ حمایت می کند.
- موتور زمان اجرای java نمونه سیستمی از یک فرآیند چند نخ است.
- در سیستمهای عامل های زیرسیستم چند نخ استفاده شده است

Windows NT ; solaris ; Mach ; os/2

موارد همراه فرآیند:

- فضای آدرس مجازی که تصویر فرآیند را در بر دارد.
- دسترسی حفاظت شده به فرآیند.
- حالت اجرای نخ.
- متن ذخیره شده.
- پشته اجرا.
- مقداری حافظه ایستا برای متغیرهای محلی هر نخ.
- دسترسی به منابع و حافظه فرآیند.

مقایسه فرآیند تک نخ و چند نخ:

- فرآیند چند نخ:
- یک بلوک کنترل فرآیند، فضای آدرس فرآیند، برای هر نخ پشته جدا گانه و ... وجود دارد.
- تمام نخ های فرآیند، در حالت و منابع آن فرآیند شریک هستند.
- در یک فضای آدرس هستند و به داده های یکسانی دسترسی دارند.

- فرآیند تک نخ:
- شامل بلوک کنترل، فضای آدرس فرآیند، فضای آدرس فرآیند، پشته کاربر، هسته برای مدیریت و ... است.

حالات نخ :

- ایجاد (زایش):
- با ایجاد یک فرآیند، نخ هم ایجاد می شود.
- مسدود شدن:
- تا بروز حادثه نخ مسدود می شود.
- رفع مسدود بودن:
- اتفاق حادثه لازم.
- پایان:
- با تکمیل یک نخ، متن ثبات و پشته هایش آزاد می شود.

همگام سازی نخ ها

- تمام نخ ها یک فرآیند در فضای آدرس و در منابع دیگر شریک هستند به همین تغییر در هر یک از منابع توسط یک نخ در محیط سایرین تغییر ایجاد می کند لذا همگام سازی فرآیند ها ضروری است.
- به دو دلیل از نخ جداگانه برای رسم مجدد صفحه نمایش استفاده می کنند:
- ۱- page maker تعدادش هایی که در روی یک صفحه است را محدود نکند.

۲- استفاده از نخ جداگانه، امکان قطع کردن رسم را در اختیار می گذارد.

نخهای سطح کاربر

- کار مدیریت توسط کاربر انجام می شود.
- هسته از وجود این نخ ها بی اطلاع است.
- از کتابخانه ها برای مدیریت نخها استفاده می کنند.
- هر کاربردی با یک نخ شروع می شود و شروع اجرایش از همان نخ است.
- هرگاه این نخ در حال اجرا باشد می توان نخ جدیدی را ایجاد کرد که به احضار کتابخانه منجر می شود.
- الگوریتم زمانبندی آن ایجاد می شود تا آماده اجرا می شود.

نخهای سطح هسته

- تمام کار مدیریت نخ توسط هسته انجام می شود.
- کد مدیریت در ناحیه کاربرد وجود ندارد.
- یک نخ واسط برنامه سازی کاربردی به نخ هسته وجود دارد.

امتیاز نخ سطح کاربر به سطح هسته:

- تعویض نخ به حالت ممتاز نیاز ندارد.
- کاربرد می تواند زمانبندی خاص داشته باشد
- نخ های سطح کاربر می توانند روی هر سیستم عامل اجرا گردند.

اشکال نخ های سطح کاربر به سطح هسته:

- اکثر فراخوانی های سیستم مسدود کننده هستند.
- کاربرد چند نخ نمی تواند از امتیازات چند پردازشی استفاده کند.

رفع اشکال نخ های سطح کاربر به سطح هسته

- نوشتن کاربردها به صورت فرآیندهای متعدد به جای نخ های متعدد که امتیاز اصلی نخ ها را از بین می برد.
- راه حل غلبه بر نخ های مسدود کننده : استفاده از روشی به نام جلد کردن که هدف آن تبدیل فراخوانی سیستم از مسدود کننده به غیر مسدود کننده.

امتیاز نخ های سطح هسته:

- (تمام کارمدیریت نخ توسط هسته انجام می شود.)
- هسته می تواند به صورت هم زمان، نخ های چند گانه یک فرآیند واحد را روی پردازنده های متعدد زمانبندی کند.
- اگر نخ در یک فرآیند مسدود شد، هسته می تواند نخ دیگری از آن فرآیند را بار گزاری کند.
- خود روالهای هسته نیز می تواند چند نخ باشد.

عیب اصلی نخ های سطح هسته:

انتقال کنترل از یک نخ به نخ دیگر در داخل یک فرآیند واحد نیز نیازمند تغییر به حالت هسته است.

رویکردهای ترکیبی:

- ترکیبی از امکانات نخ سطح کاربر و نخ سطح هسته را فراهم می کند. مانند solaris .
- ایجاد نخ به طور کامل در فضای کاربرد در داخل یک کاربرد صورت می گیرد.
- به تعداد (کمتر یا مساوی) از نخ های سطح هسته نگاشته می شود.
- نخ ها به موازات هم روی پردازنده های متعدد اجرا می شود.
- یک فراخوان مسدود کننده کل سیستم را مسدود نمی کند.

رابط بین نخ ها و فرآیندها:

نخها: فرآیندها	شرح	مثال سیستم های عامل
یک به یک	هر نخ اجرا یک فرآیند	unix
چند به یک	یک فرآیند یک فضای آدرس	windows N T
یک به چند	یک نخ از یک محیط به دیگر رود	Ra(clouds)
چند به چند	ترکیب چند به یک و یک به چند	TRIX

روش های پیاده سازی رابط چند به چند:

- تمام برنامه به صورت یک فرآیند واحد پیاده سازی شود.
- از نظر مدیریت حافظه اشکالاتی دارد.
- برنامه اصلی و زیر برنامه ورودی / خروجی به صورت دو فرآیند جداگانه پیاده سازی شوند. مستلزم سر بار اضافی است.
- برنامه اصلی و زیر برنامه ورودی / خروجی به عنوان یک فعالیت و به صورت یک نخ پیاده سازی شود. ولی یک فضای آدرس و یکی برای زیر برنامه ورودی / خروجی ایجاد گردد. (بهترین روش است).

معماری چند پردازشی متقارن:

FLYNN گروه های زیر را پیشنهاد کرده است:

- جریان دستورالعمل واحد و داده های واحد: یک پردازنده دستورات واحدی را برای داده ها دریافت می کند.
- جریان دستورالعمل چند گانه و داده های واحد: اجرای هم زمان تعدادی از دستورات پردازشی.
- جریان دستورالعمل چند گانه و داده های چند گانه: اجرای هم زمان چند دستور بر روی چند داده.
- جریان دستورالعمل واحد و داده های چند گانه: اجرای هم زمان چند دستور بر روی داده.

سازمان چند پردازنده متقارن:

- هر يك واحد كنترل ،واحد حساب و منطق و ثباتهای خاص خود را دارند.
- به يك حافظه اصلی مشترك دسترسی دارد.
- پردازنده ها از طریق حافظه با يكديگر در ارتباط هستند.
- امکان تبادل مستقیم وجود دارد.
- هر پردازنده از طریق يك گذرگاه مشترك به يك حافظه اصلی متصل است.
- امکان دسترسی به بلوك ها مهيا است.
- تغییر در يك حافظه پنهان سبب تغییر در حافظه پنهان دیگر می شود.(وابستگی حافظه پنهان)
- با آن در سخت افزار بر خورد می شود.

ملاحظات طراحی سیستم عامل چند پردازنده ای:

- هم زمانی فرآیند ها یا نخ ها: توسط هر يك از پردازنده ها انجام شود .
- زمانبندی: توسط هر يك از پردازنده ها انجام شود .
- همگام سازی: ابزاری است برای تنظیم انحصار متقابل و تنظیم حادثه ها.
- مدیریت حافظه: باید سخت افزار مناسب را بهره جوید.
- قابلیت اطمینان و تحمل خرابی: در برخورد با خرابی با ید اقدام کند.

ریز هسته ها:

- ریز هسته سیستم عامل کوچکی است که بنیان لازم برای گسترشهای مولفه ای را به وجود می آورد.
- رویکرد ریز هسته با به کارگیری آن در سیستم عامل Mach مورد توجه عمومی قرار گرفت.
- سیستم عامل windows NT نیز دارای ریز هسته است که مدعی قابلیت حمل و مولفه ای بودن است.

معماری ریز هسته:

- در این سیستم های عامل یکپارچه، مجازاً هر رویه ای می توانست هر رویه دیگری را فراخوانی کنند.
- در سیستم عامل های لایه ای،عمل ها به صورت سلسله مراتبی طراحی شده است و تنها سطوح مجاور با یکدیگر محاوره داشته باشند.
- فلسفه ریز هسته این است که تنها اعمال کاملاً اصلی سیستم عامل باید در هسته باشد.

- کاربردها روی ریز هسته ساخته شده.
- بخشهایی از سیستم عامل که در خارج ریز هسته قرار دارند به صورت فرآیند خدمتگذار پیاده سازی می شوند.
- بر مبنای برابری و از طریق ریز هسته با هم ارتباط برقرار می کنند.
- ریز هسته به عنوان یک مبادله کننده پیام عمل می کند.

مزایای ریز هسته:

- واسط یکنواخت. (برای درخواستهای فرآیند)
- قابلیت گسترش. (مجاز دانستن افزایش خدمات)
- قابلیت انعطاف. (به قابلیت گسترش مربوط است)
- قابلیت حمل. (تمام کد مربوط به ویژگی های خاص پردازنده در هسته وجود دارد)
- قابلیت اطمینان.
- حمایت سیستم توزیعی: (توسط سیستم عامل توزیعی)
- سیستم عامل شیء گرا.

کارایی ریز هسته:

- ریز هسته خیلی کوچک سبب شده که کاهش کارایی را حذف و انعطاف و قابلیت اطمینان را بهتر کند.
- به دلیل اینکه ریز هسته های مختلف طیفی از اعمال و اندازه را نشان می دهند، نمی توان هیچ قاعده محکم و سریعی در مورد اعمالی که ریز هسته باید ارائه کند و ساختار پیاده سازی آن اظهار داشت.

طرح ریز هسته:

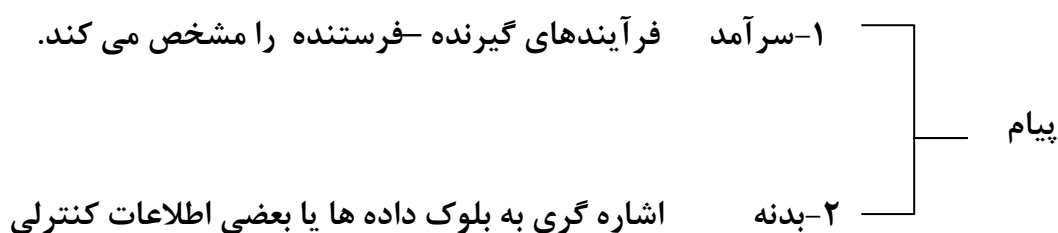
- ریز هسته مستقیماً به سخت افزار بستگی دارد و اعمال زیر را حمایت می کند:
 - ۱- مدیریت حافظه اولیه.
 - ۲- ارتباط بین فرآیندها.
 - ۳- ورودی / خروجی.
 - ۴- مدیریت وقفه.

سه روش زیر برای حمایت مدیریت حافظه مجازی و صفحه بندی خارجی:

- اهدا: هر فرآیند می تواند تعدادی از صفحات خود را به دیگری دهد.
- نگاشت: یک فرآیند می تواند هر یک از صفحات خود را به فضای دیگری بنگارد.
- مطالبه: هر صفحه که قبلا استفاده شده را مطالبه کند.

ارتباط داخلی فرآیند

پیام شکل اصلی ارتباط بین فرایندها یا نخهاست.



مدیریت ورودی و خروجی و وقفه ها:

- اداره وقفه های سخت افزاری به صورت پیام و درگاه های ورودی و خروجی در فضاهای آدرس میسر می شود.
- ریز هسته وقفه را تشخیص می دهد ولی آنها را اداره نمی کند.
- پیامی را برای فرآیند سطح کاربر مربوط به آن وقفه تولید می کند.

مدیریت نخ و چند پر دازشی متقارن در

Windows NT

ویژگیهای فرآیندهای NT عبارتند از:

- فرآیند NT به صورت شیء پیاده سازی شده اند.
- هر فرآیند قابل اجرا می تواند حاوی یک نخ یا بیشتر باشد.
- هم شیء های فرآیند و هم شیء های نخ، دارای راهکارهای پیش ساخته همگام سازی هستند.

- به هر فرآیند مجموعه بلوک های حافظه مربوط است.
- به هر فرآیند یک نشانه امنیتی دست یابی نسبت داده می شود.
- فرآیند شامل یک جدول شیء هم هست که دستگیره های به دیگری شیء های شناخته شده این فرایند دارد. برای هر یک از نخهای که در این شیء هست یک دستگیره وجود دارد.
- اگر فرآیندی برای باز کردن چنین دستگیره های تلاش کند، سیستم امنیتی مجاز بودن آن و در نتیجه امکان تغییر خصوصیاتش را تعیین می کند.
- فرآیند نمی تواند مستقیما ساختار ها را تغییر دهد.
- فرایند شامل یک جدول شیء است .

شناسه فرآیند	مقدار یکتای فرآیند
توصیف امنیت	توصیف چه کسی به این فرآیند...
اولویت پایه	برای نخ های این فرآیند
وابستگی فرآیند به پردازنده	مجموعه پردازنده های پیش فرض
حداکثر سهم	حداکثر منابع
زمان اجرا	مجموع زمان اجرا نخ
I/O شمارنده های	متغیر و... ثبت شده توسط فرآیند
شمارنده عملیات حافظه مجازی	تعداد و نوع عملیات انجام شده
درگاه های استثنا/اشکالزدایی	کانالهای ارتباطی بین فرایندها
وضعیت خروج	دلیل پایان یافتن یک فرآیند

خصوصیت شیء نخ :

- شناسه نخ: مقدار یکتای نخ.
- متن نخ: مجموع مقادیر نخ و....
- اولویت پویا: اولویت اجرای نخ در هر لحظه.
- اولویت پایه: کمترین اولویت نخ.
- وابستگی نخ به پردازنده: پردازنده هایی که می توان در روی نخ انجام شود.
- زمان اجرای نخ: زمان اجرای آن در هسته و در حالت کاربرد.
- وضعیت هشدار: فراخوانی رویه نا همگام .
- شماره تعلیق : تعداد دفعات تعلیق اجرای نخ.
- نشانه جعل هویت: دستیابی موقت .
- درگاه پایان: یک کانال ارتباطی بین فرآیندها.
- وضعیت خروجی نخ: دلیل پایان یافتن نخ.

حالت های نخ:

- آماده: برای اجرا زمانبندی گردیده.
- جانشین: برای هر پردازنده خاص ،یک نخ جانشین
- اجرا: با تعویض به مرحله اجرا می رود.
- انتظار: یک نخ موقعی به حالت اجرا می رود که:
- ۱- روی حادثه مسدود گردد.
- ۲- به منظور همگام سازی ،داوطلبانه منتظر شود.
- ۳- توسط یک زیر سیستم محیط وادار به تعلیق گردد.
- انتقال: منابع فراهم نیست.
- پایان: توسط خودش یا پدر.

مدیریت نخ و چند پردازشی متقارن در Solaris

- فرآیند: همان فرآیند معمولی

- نخ های سطح کاربر: به وسیله کتابخانه ای از نخها در فضای آدرس یک فرآیند پیاده سازی می شوند.
- فرآیند سبک وزن: نگاشتی بین نخ های سطح کاربر و نخ های هسته در نظر گرفت.
- نخ های هسته: موجودیتهای بنیادی.

اجرای نخ:

- همگام سازی: جهت هما هنگ کردن فعالیت.
- تعلیق : معلق شدن و منتظر درخواست.
- قبضه : عملی که موجب به اجرا در آمدن فعالیت اولویت دار شود.
- واگذاری.

راه حل solaris برای اداره وقفه

- برای اداره وقفه ها solaris مجموعه ای نخ های هسته را به کار گرفته است.
- با استفاده از اولیه های انحصار متقابل ،هسته دسترسی به ساختمان داده ها را کنترل می کند.
- اولویت وقفه ها ،از اولویت دیگر نخ های هسته بیشتر است.

فصل پنجم

همزمانی:

انحصار متقابل و همگام
سازی

همه موضوعات محوری در طراحی سیستم عامل به مدیریت فرایند ها و نخ ها مربوط است.

- چند برنامه ای : مدیریت فرایند فرایندهای متعدد در داخل یک کامپیوتر تک پردازنده ای.
- چند پردازشی: مدیریت فرایند فرایندهای متعدد در داخل یک کامپیوتر چند پردازنده ای.
- پردازش توزیعی: مدیریت فرایند فرایندهای متعدد در روی سیستم های کامپیوتری متعدد.

برای هر سه زمینه فوق مسئله هم زمانی است.

همزمانی در سه زمینه متفاوت طراحی می گردد:

- کاربردهای متعدد: زمان پردازش کامپیوتر به طور پویا بین تعدادی کار یا کاربرد بتواند تقسیم گردد.
- کاربرد ساختیافته: بعضی از کاربردها رابه صورت مجموعه ای از فرایندهای همزمان، به طور کارآمد برنامه سازی کرد.
- ساختار سیستم عامل: همین امتیاز ساختاردهی، برای برنامه ساز سیستم نیز قابل اعمال است.

نیاز اصلی برای حمایت از فرآیندهای همزمان، توان اعمال انحصار متقابل است.

یعنی وقتی به یک فرآیند قدرت انجام عملی داده شد. بتوان تمام فرایندها دیگر را از این قدرت بازداشت.

اصول همزمانی

در سیستم های چند پردازنده ای، نه تنها ممکن است فرایند ها در بین یکدیگر اجرا شوند، بلکه می توانند واقعا به موازات هم و با همپوشانی اجرا گردند.

در این روش مشکلات زیر پیش می آید:

- ۱- اشتراک منابع سراسری پر مخاطره است.
- ۲- مدیریت تخصیص بهینه منابع برای سیستم عامل است
- ۳- تعیین محل خطای برنامه سازی مشکل می شود.

ملاحظات سیستم عامل در همزمانی:

- سیستم عامل باید به تواند فرایندهای فعال مختلف را دنبال کند که این کار توسط بلوک های کنترل فرایند انجام می شود.

- سیستم عامل باید منابع را به هر یک از فرایندها تخصیص دهد و یا باز پس بگیرد از جمله:

(وقت پردازنده، حافظه، پرونده هاریا، دستگاههای ورودی و خروجی)

- سیستم عامل باید داده ها و منابع فیزیکی هر فرایند را در مقابل دخالت نا خواسته فرایند دیگر محافظت کند.

- نتایج یک فرایند باید مستقل از سرعت پیشرفت اجرای فرایندهای همزمان دیگر باشد.

محاوره فرایندها:

محاوره فرآیندها: ارتباطات بین فرآیندها و نخ ها

- بی اطلاعی فرایندها از یکدیگر: فرایندها مستقل هستند.
- اطلاع غیر مستقیم فرایندها از یکدیگر: دسترسی به بعضی از اشیاء یکدیگر.
- اطلاع از یکدیگر: با نام ، با یکدیگر در ارتباط هستند.

هنگامی که چند فرایند برای استفاده از یک منابع فعالیت می کنند درگیری دارند که

شرایط اساسی در راه حل صحیح:

- انحصار متقابل در اجرای بخش بحرانی.
- فعالیت های محدوده زمانی در هر بخش.
- تعداد مرتبه های رفتن به ناحیه بحرانی در حضور منتظران محدود است.
- هیچ فرضی نسبت به طول دستورات و سرعت اجرا نمی باشد.
- فرایند: حلقه همیشگی.
- در پایان بن بست به باقی فرایندها پیامی ارسال می گردد.
- گرسنگی.

همکاری فرایندها توسط اشتراک:

- فرایندهایی را پوشش می دهند که با یکدیگر محاوره می کنند، بدون اینکه صراحتاً از یکدیگر مطلع باشند.

- مسئله کنترل انحصار متقابل ، بن بست ، گرسنگی باز وجود دارد و داده در دو حالت ممکن خواندن و نوشتن مورد دسترسی قرار می گیرند و تنها مورد نوشتن باید در انحصار متقابل قرار گیرد.

همکاری فرایند ها توسط ارتباط:

- هر فرایند محیط مجزای خود را دارد و فرایند های دیگر را در بر ندارد.
- محاوره بین فرایند ها غیر مستقیم است و فرایند ها صراحتا از محل یکدیگر اطلاع ندارند.
- ارتباط را با نوعی پیام مشخص می کنند.
- مسئله بن بست و گرسنگی وجود دارد.

ملزومات انحصار متقابل:

- انحصار متقابل باید اعمال گردد:
- فرایندی که در بخش غیر بحرانی خود متوقف می شود ، باید طوری عمل کند که هیچ دخالتی در فرایند های دیگر نداشته باشد .
- برای فرایندی که نیاز به دسترسی یک بخش بحرانی دارد،نباید به تاخیر انداختن نامحدود آن وجود داشته باشد ، بن بست یا گرسنگی نمی تواند مجاز باشد.
- هنگامی که هیچ فرایندی در ناحیه بحرانی نیست ورود یک فرایند بدون تاخیر است.
- هیچ فرضی در باره تعداد فرایند ها و یا سرعت آن داشت.
- هر فرایندی تنها برای زمان محدودی در ناحیه بحرانی قرار می گیرد.

انحصار متقابل: رویکرد نرم افزاری

رویکرد نرم افزاری را میتوان برای فرایند های همزمانی که روی ماشینهای تک پردازنده ای یا چند پردازنده ای که از حافظه مشترک استفاده می کنند پیاده سازی کرد.

الگوریتم DEKKER:

برای انحصار متقابل دو فرایند ،الگوریتمی را منتشر کرد .
امتیاز این رویکرد آن است که حداکثر خطاهای متداول در ایجاد برنامه های همزمان ،تشریح می شود.

الگوریتم Peterson :

آرایه سراسری flag نمایانگر وضع هر فرایند نسبت به انحصار متقابل است و متغیر سراسری turn درگیریهایی همزمانی را حل می کند.

انحصار متقابل: حمایت سخت افزار

از کار انداختن وقفه:

- در یک ماشین تک پردازنده، هم زمان نمی شود فرایند ها وارد ناحیه بحرانی شوند.
- برای ضمانت انحصار متقابل، کافی است از مواجهه فرایند با وقفه جلوگیری شود.
- چون بخش بحرانی نمی تواند وقفه داده شود، لذا انحصار متقابل ضمانت شده است.
- کارایی اجرایی به طور متقابل کم می شود.
- این روش در معماری چند پردازندهای کارایی ندارد.

دستور العمل آزمون و مقدار گذاری:

- در این روش هر فرایند در هنگام وارد شدن به ناحیه بحرانی مقدار متغیر را یک می کند و هنگام خارج شدن آن را صفر می کند، هنگامی که مقدار یک می باشد هیچ فرایندی اجازه ورود به این ناحیه را ندارد.

ویژگیهای رویکرد دستورالعمل ماشین:

- برای هر تعداد از فرایندها، روی یک پردازنده و یا چند پردازنده که از حافظه مشترک استفاده می کنند، قابل به کارگیری است.
- ساده است و لذا واریسی آن آسان می باشد.
- از آن برای حمایت از بخشهای بحرانی متعدد می توان استفاده کرد.

معایب رویکرد دستورالعمل ماشین:

- انتظار مشغول جود دارد.
- امکان گرسنگی وجود دارد.

راهنماها:

- برای علامت دهی بین فرایند ها و اعمال یک نظام انحصار متقابل استفاده کرد.

- اصل اساسی این است که دو یا چند فرایند بتوانند به وسیله علائم ساده ای با یکدیگر همکاری کنند.
- برای علامت دادن از متغییر ویژه ای به نام راهنما استفاده می کنند.
- برای فرستادن علامت توسط راهنمای S، فرایند اولیه (s) signal را اجرا می کند.
- برای دریافت علامت توسط یک راهنمای S، فرایند اولیه wait(s) را اجرا می کند.
- اگر علامت مربوط ، هنوز فرستاده نشده باشد ،این فرایند تا زمان ان،معلق می ماند.

ناظرها:

- ساختاری از زبان برنامه سازی است که کار راهنما را انجام می دهد و کنترل آن هم ساده تر است.
- ساختار ناظر در تعدادی از زبانهای برنامه سازی از modula و ... وبه صورت کتابخانه ای پیاده سازی می شود.
- این افراد اجازه می دهد تا قفل های ناظر در روی هر شیء بگذارد.
- در عمل مانند لیست پیوندی است.

ناظر با علامت:

مولفه ای نرم افزاری مشتمل بر یک یا چند رویه، دنباله ای از مقدار گذاری اولیه و داده های محلی است .

دارای ویژگی های :

۱. متغیرهای داده محلی ناظر تنها برای رویه های خود ناظر قابل دسترسی بوده و هیچ رویه دیگری به آنها دسترسی ندارد .
۲. یک فرایند با احضار یکی از رویه های ناظر ،وارد ان می شود.
۳. در هر زمان تنها یک فرایند می تواند در ناظر در حال اجرا باشد.

دو تابعی که روی متغییر شرطی عمل می کند عبارتند از:

Cwait(c) : اجرای یک فرایند صدا کننده را روی شرط C معلق می کند.

Csignal(c) : اجرای یک فرایندی که بعد از یک عمل cwait ، روی همان شرط معلق بوده است را از سر می گیرد.

تبادل پیام:

- پیام ها برای انحصار متقابل مفید هستند.
- ابزاری موثری را برای ارتباط بین فرآیندها فراهم می کند.

- هنگامی که دو فرآیند با یکدیگر ارتباط برقرار می کند، دو نیاز بنیادی همگام سازی و ارتباط باید تامین شود.
- فرایندها نیاز به همگام سازی دارند تا انحصار متقابل تامین شوند.
- Send (destination,message)
- Receive(source,message)

همگام سازی

در هنگام تبادل پیام سه ترکیب زیر متداول است :

- مسدود شدن فرستنده ،مسدود شدن گیرنده:

گاهی به این قرار ملاقات میگویند.همگام سازی بین فرایندها را محکم می کند.

- مسدود نشدن فرستنده ،مسدود شدن گیرنده:

گیرنده تا زمان ورود پیام مسدود است.

- مسدود نشدن فرستنده ، مسدود نشدن گیرنده .

آدرس دهی:

- آدرس دهی مستقیم:

۱. Send شامل شناسه آدرس فرایند مقصد.

۲. receive یا صریح بیان می کند یا ضمنی.

- آدرس دهی غیر مستقیم:

پیام ها مستقیماً از فرستنده به گیرنده فرستاده نمی شوند بلکه به یک ساختمان داده مشترک ارسال میگردد.

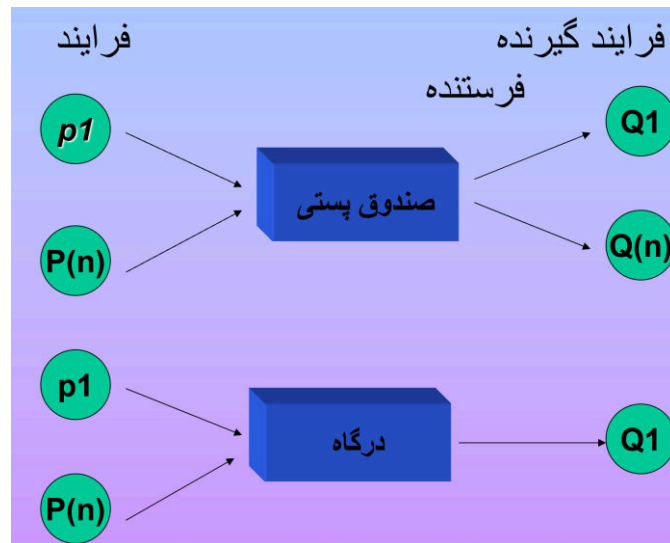
این ساختمان داده دارای صفهایی است که به آن ها صندوق پستی می گویند.

انتساب پیام به صندوق پستی می تواند : پویا و ایستا باشد.

ایستا: به یک فرایند نسبت داده می شود.

پویا: هنگامی که فرستنده های زیادی وجود داشته باشد.

ارتباط غیر مستقیم فرایندها:



قالب پیام:

- قالب پیام شامل:
 ۱. سرآمد: حاوی اطلاعات درباره پیام.
 ۲. بدنه: حاوی خود پیام.



نظام صف بندی

- FIFO ساده ترین نظام صف بندی
- اولیت پیام بر اساس نوع پیام
- اولیت پیام توسط فرستنده
- اولیت پیام توسط گیرنده (بازرسی صف پیامها و انتخاب پیام بعدی)

وضعیت صفهای فرایندها :

برقراری wsem بدون صف	وجود فقط خوانندگان در سیستم
برقراری wsem و rsem صف نویسندگان روی wsem	وجود فقط نویسندگان در سیستم
برقراری wsem به وسیله خواننده برقراری rsem به وسیله نویسنده صف تمام نویسندگان روی wsem صف تنها یک خواننده روی rsem صف خوانندگان دیگر روی Z	هم خوانندگان و هم نویسندگان با تقدم خواندن
برقراری wsem به وسیله خواننده برقراری rsem به وسیله نویسنده صف تمام نویسندگان روی wsem صف تنها یک خواننده روی rsem صف خوانندگان دیگر روی Z	هم خوانندگان و هم نویسندگان با تقدم نوشتن:

فصل ششم

همزمانی: بن بست و

گرسنگی

اصول بن بست:

- بن بست را به صورت مسدود بودن دائمی مجموعه ای از فرآیند ها که برای منابع سیستم رقابت می کنند یا با یکدیگر در ارتباط هستند .
- راه حل کارامدی برای بن بست وجود ندارد.
- تمام بن بست ها نیاز های متضاد دو فرآیند یا بیشتر ،برای منابع هم راه هستند.

انواع منابع:

- **منابع قابل استفاده مجدد:** بدون صدمه توسط یک فرآیند می توان در هر زمان مورد استفاده قرار می گیرد و تمام نمی شود.
- مثل :پردازنده ها،حافظه اصلی و ثانویه و...
- **منابع مصرف شدنی:** منبعی هستند که می تواند ایجاد و نابود گردد.
- نوعا روی تعداد منابع مصرف شدنی از یک نوع به خصوص ،حدی وجود ندارد.

شرایط بن بست:

- **انحصار متقابل : (شرط لازم)**
- یک فرآیند از یک منبع.
- **نگهداشتن و انتظار: (شرط لازم)**
- با داشتن منبع دیگری در خواست منبع جدید می کند.
- در این صورت یک منبع به مدت طولانی در اختیار یک فرآیند خواهد بود.
- **قبضه نکردن: (شرط لازم)**
- منبع را نمی توان به زور پس گرفت.
- تنها وقتی از بن بست جلوگیری می کند :که منابع دارای اولویت باشند.
- **انتظار مدور: (شرط لازم و کافی)**
- چند فرآیند درخواست منبع هایی می کنند که در اختیار منبع دیگر است
- و به همین سبب در انتظار چرخشی قرار می گیرند.
- موجب کند کردن فرآیند ها و رد کردن غیر ضروری دسترسی منابع می شود.

سه رویکرد در برخورد با بن بست

- پیشگیری
- کشف
- اجتناب

پیشگیری از بن بست:

- روش غیر مستقیم: پیشگیری از حالت‌های ۱ و ۲ و ۳ (انحصار متقابل، نگهداشتن و انتظار، قبضه نکردن)
- روش مستقیم: پیشگیری از بروز انتظار مدور

مزایای پیشگیری از بن بست:

- در مورد فرآیند‌هایی که فعالیت‌هایی زیادی انجام می‌دهند.
- عدم نیاز به قبضه کردن.
- معایب:
- نا کارآمدی.
- تاخیر شروع فرآیند.

کشف:

- سیاست تخصیص منابع: بسیار زیاد، در صورت امکان تخصیص داده می‌شود.
- طرح‌های مختلف: احضار دورها برای بررسی بن بست.

مزایای اصلی:

- عدم تاخیر در آغاز فرآیند.
- تسهیل پردازش در حین کار.

معایب اصلی:

- ضررهای ذاتی قبضه.

اجتناب:

- سیاست تخصیص منابع: انتخاب راهی بین پیشگیری و کشف.
- طرح‌های مختلف: دستکاری برای یافتن حداقل یک مسیر امن.

مزایای اصلی:

- عدم نیاز به قبضه کردن.

معایب اصلی:

- ضرورت اطلاع از منابع مورد نیاز آینده
- امکان مسدود شدن طولانی فرآیند.

دو رویکرد برای اجتناب از بن بست

- عدم شروع فرایندی که ممکن است درخواستهایش به بن بست منتهی شود
- عدم پاسخ به درخواستهای منبع اضافی از فرایندی که با این تخصیص ممکن است منجر به بن بست شود.

عدم تخصیص:

- الگوریتم بانکداران نیز می گویند.
- حالت سیستم تخصیص منابع به فرآیند ها است.
- دارای دو حالت :
- ۱-حالت امن:
- حداقل یکی از فرآیند ها اجرا شود.
- ۲-حالت نا امن:

حالتی است که امن نباشد.

در این روش باید منابع با دقت به درخواست کننده ها ارائه شود که دچار کمبود منابع نشویم .

امتیاز اجتناب از بن بست:

- حداکثر منابع مورد نیاز هر فرآیند باید پیش بینی شود.
- فرآیندهای مورد نیاز باید مستقل باشند.
- تعداد منابع تخصیص باید ثابت باشند.
- فرآیندی که منبعی را در اختیار دارد نمی تواند خارج شود.

ترمیم : (بعد از کشف بن بست)

- قطع تمام فرآیندهای بن بست.
- برگشت هر یک از فرآیندهای بن بست به نقطه قبل از آن.

- قطع پی در پی بن بست تا جایی که بن بست وجود نداشته باشد.
- قبضه کردن منابع تا جایی که بن بست وجود نداشته باشد.

راهبرد مجتمع برای بن بست:

- تقسیم بندی منابع در تعدادی از گروههای مختلف.
- استفاده از راهبرد مرتب سازی خطی.
- استفاده از مناسب ترین الگوریتم.

به عنوان مثال استفاده از بهترین الگوریتم:

- فضای قابل مبادله: بلوک هایی از حافظه ثانویه برای استفاده در مبادله فرآیند.
- منابع فرآیند: دستگاههای قابل تخصیص.
- حافظه اصلی: حافظه قابل تخصیص به فرآیندها.
- منابع داخلی: مثل کانالهای ورودی و خروجی.

مسئله تغذیه فیلسوفان:

- هر فیلسوفی که می خواست غذا بخورد به جایی که در این میز منظور شده بود می رفت و با استفاده از دو چنگال کنار بشقابش غذا می کشد مشکل این است که باید چاره ای برای رفع گرسنگی سایرین نیز داشته باشیم.
- برای این مشکل هر فیلسوف ابتدا چنگال سمت چپ و سپس چنگال سمت راست خود را بردارد که باز هم در این روش مسئله بن بست پیش می آید.
- مسائل اصلی بن بست گرسنگی را تشریح می کند.

راهکارهای هم زمانی در unix :

- لوله ها
- پیام ها
- حافظه مشترک
- راهنماها

علائم

- لوله ها:

یکی از مهم ترین کمکهای unix به ایجاد و توسعه سیستم های عامل است.
بر اساس مدل تولید کننده و مصرف کننده با هم ارتباط برقرار می کند.

صف خروجی به ترتیب ورودی است. دو نوع لوله بی نام و با نام وجود دارد.

• پیام ها:

بلوکی از متن به همراه یک نوع است. از فراخوان سیستمی msgsnd و msgrcv برای فرآیندهای درگیر در تبادل پیام استفاده می کنند.

• حافظه مشترک:

سریعترین شکل ارتباط بین فرآیندها است. بلوکی از حافظه مجازی است که توسط فرآیندهای متعدد مورد اشتراک قرار می گیرد .

• راهنماها:

از اولیه های wait و signal است. عملیات متعددی به صورت هم زمان اجرا می شود. یک راهنما شامل عناصر زیر است:

۱- مقدار جاری راهنما.

۲- شناسه.

۳- تعداد فرآیندهای منتظر مقدار راهنما از مقدار جاری آنها بزرگتر باشد.

۴- تعداد فرآیندهای منتظر مقدار راهنما ی آنها صفر باشد...

• علائم:

یک راهکار نرم افزاری است که از بروز یک حادثه نا بهنگام خبر می دهد.

اولیه های همگام سازی نخ در solaris:

• قفل های انحصار متقابل:

از پیش رفت یک نخ جلوگیری می کند.

• راهنماها:

راهنماهای شمارنده کلاسیک را ارائه می دهد.

• قفل های چند خواننده و یک نویسنده:

اجازه می دهد نخ های متعدد هم زمان به شیء که تحت حفاظت است دسترسی فقط خواندنی داشته باشد.

- متغیرهای شرط:

برای انتظار تا برقراری شرط خاصی.

راهکارهای همزمانی در windows NT:

- فرآیند: احضار یک برنامه .
- نخ: موجودیت قابل اجرا .
- پرونده: نمونه پرونده باز.
- ورودی میز کار: یک میانگیر صفحه نمایش متنی.
- اعلام تغییر پرونده: تغییر سیستم پرونده.
- Mutex (انحصار متقابل).
- راهنما: شمارنده تعداد رخ های برنامه را تنظیم می کند.
- حادثه: یک حادثه سیستمی اتفاق افتاده است.
- زمان سنج قابل انتظار: شمارنده برای ثبت گذشت زمان.

فصل هفتم

مدیریت حافظه

نیازهای مدیریت حافظه:

دارای پنج نیاز زیر است:

- **جابجایی:** برای این منظور سیستم باید از محل فرآیند ها آگاهی داشته باشند و آدرس آنها را ذخیره کنند.
- **حفاظت:** باید در مقابل تداخل های ناخواسته حفاظت شود خواه تصادفی باشد یا عمدی .
- **اشتراک:** باید با داشتن حفاظت به گونه باشد تا دیگران نیز بتوانند به آن دستیابی داشته باشند .
- **سازمان منطقی:** حافظه به صورت فضای آدرس خطی یا یک بعدی سازمان یافته است و شامل دنباله ای از بایتها و کلمه ها است .
- **سازمان فیزیکی:** حافظه به دو صورت اصلی که ناپایدار ،سریع ،گران و مجازی که پایدار ،نسبتا ارزان می باشد و مدیریت بین این دو می باشد.

مزایای سازمان دهی منطقی:

- هر مولفه را می توان به صورت مستقل نوشت و ترجمه کرد.
- با یک سر بار اضافی مراتب مختلف حفاظتی را داریم.
- امکان معرفی راهکارهایی برای اشتراک مولفه ها در بین فرآیند ها وجود دارد.

علت نا کارآمدی سازمان دهی فیزیکی :

- ممکن است حافظه موجود برای داده ها کافی نباشد.
- در یک محیط چند برنامه،برنامه سازدر زمان نوشتن برنامه نمی داند چه مقدار حافظه در اختیار دارد و...
- **روش های مدیریت حافظه:** بخش بندی ایستا: حافظه اصلی به تعدادی بخش ایستا در زمان ایجاد سیستم ایجاد می شود.فرآیند به داخل بخشی با اندازه برابر یا بزرگتر برود.
- **بخش پویا:** بخشها به صورت پویا ایجاد می شود،هر فرآیند به داخل بخشی برابر با اندازه خودش می رود.
- **صفحه بندی ساده:** حافظه به قاب هایی هم اندازه تقسیم می شودو فرآیند به صفحات هم اندازه با قاب ها تقسیم می شود.

- **قطعه بندی ساده:** هر فرآیند به قطعه هایی تقسیم می شود و از طریق باز کردن تمام قطعاتش اجرا می شود که لزوما پیوسته نیستند.
- **صفحه بندی مجازی:** مانند صفحه بندی ساده است ولی نیاز نیست تمام صفحات یک فرآیند باز شود.
- **قطعه بندی حافظه مجازی:** مانند قطعه بندی ساده است ولی نیاز نیست تمام صفحات یک فرآیند باز شود.

مزایا:

- **بخش بندی ایستا:** سادگی پیاده سازی، کمی سر بار سیستم.
- **بخش بندی پویا:** بدون تکه تکه شدن داخلی.
- **صفحه بندی ساده:** بدون تکه تکه شدن خارجی.
- **قطعه بندی ساده:** بدون تکه تکه شدن داخلی.
- **صفحه بندی حافظه مجازی:** بدون تکه تکه شدن داخلی، درجه چند برنامه‌گی بالاتر، فضای مجازی بزرگ.
- **قطعه بندی حافظه مجازی:** بدون تکه تکه شدن داخلی، درجه چند برنامه‌گی بالاتر، فضای آدرس مجازی بزرگ، حمایت از اشتراک و حفاظت.

معایب:

- **بخش بندی ایستا:** استفاده غیر موثر از حافظه.
- **بخش بندی پویا:** استفاده غیر موثر از پردازنده.
- **صفحه بندی ساده:** مقدار کمی تکه تکه شدن داخلی.
- **قطعه بندی ساده:** گسترش به کار گیری حافظه و کاهش سربار نسبت به بخش بندی پویا.
- **صفحه بندی حافظه مجازی:** سربار پیچیدگی مدیریت حافظه.
- **قطعه بندی حافظه مجازی:** سربار پیچیدگی مدیریت حافظه.

تکه تکه داخلی:

- اینکه قسمتی از داخل یک بخش به هدر رود پدیده تکه تکه شدن داخلی می گویند.

صفحه بندی:

- حافظه اصلی به بخشهای نسبتاً کوچک تقسیم می شود.
- هر فرآیند به تکه های هم اندازه با آنها تقسیم می شود.
- به تکه های هر فرآیند صفحه می گویند.
- به تکه های موجود در حافظه قاب گویند.
- سیستم عامل یک جدول صفحه برای هر فرآیند ایجاد می کند.
- تبدیل آدرس منطقی به فیزیکی توسط سخت افزار انجام می شود.

مراحل زیر برای ترجمه آدرس لازم است:

- استخراج شماره صفحه به عنوان n بیت چپ آدرس منطقی.
- به کارگیری شماره صفحه به عنوان شاخص به جدول صفحه برای یافتن شماره قاب (k) .
- آدرس فیزیکی شروع قاب.

قطعه بندی:

- روش جایگزین برای تقسیم برنامه کاربر، قطعه بندی است.
- داده ها به تعدادی قطعه تقسیم می شود.
- قطعه دارای اندازه یکسان می باشد.
- مشابه بخش بندی پویا است.
- قطعه بندی قابل رویت است.
- رابط سندهای بین آدرس فیزیکی و منطقی وجود ندارد.

مراحل ترجمه آدرس:

- استخراج شماره قطعه از n بیت سمت چپ آدرس منطقی.
- استفاده از شماره قطعه به عنوان شاخص.
- مقایسه انحراف موجود در m بیت سمت راست.
- آدرس فیزیکی مورد نظر عبارتست از مجموع آدرس فیزیکی شروع قطعه و انحراف.

فصل هشتم

حافظه مجازی

مشخصه صفحه بندی و قطعه بندی:

- کلیه مراجع یک فرآیند به حافظه ، آدرس منطقی هستند که به آدرس فیزیکی تبدیل می شوند.
- ممکن است یک برنامه به تکه های مختلفی شکسته شود و لازم نیست این تکه ها در حین اجرا به طور پیوسته در حافظه قرار گیرد.
- تکه به صفحه یا قطعه اطلاق می شود.
- به بخشی از فرآیند که مقیم در حافظه است را مجموعه مقیم گویند.
- برای استفاده از این مجموعه از جدول صفحه استفاده می کنند.
- اگر فرآیند به آدرس منطقی دسترسی نداشته باشد یک وقفه صادر می کند.
- این وقفه به معنی خطای دسترسی به حافظه وجود دارد و فرآیند مسدود می شود و برای اجرا ان فرآیند سیستم عامل ان را به حافظه اصلی می برد.

دو پیامد این امر:

- فرآیند بیشتری را می توان در داخل حافظه اصلی قرار داد و استفاده بهتر از پردازنده می شود.
 - امکان دارد یک فرآیند بزرگتر از حافظه اصلی باشد.
- لذا برنامه نویس باید از اندازه فرآیند ها آگاه باشد.
- با وجود صفحه بندی و قطعه بندی این وظیفه به عهده سخت افزار و سیستم عامل است.

حافظه حقیقی و مجازی

حافظه حقیقی : به حافظه اصلی حافظه حقیقی می گویند.

حافظه مجازی : حافظه بزرگتر و کارآمد تر که کاربر از آن استفاده می کند حافظه مجازی می گویند.

کوبیدگی:

- سیستم عامل برای آن که تعداد بیشتری فرآیند را اجرا کند فرآیند ها را تکه تکه کرده و تکه آماده اجرا را در حافظه اصلی قرار می دهد اما ممکن است تکه ای که از حافظه بیرون رفته دوباره به خواهد اجرا شود که سیستم عامل مجبور به باز کردن دوباره آن می شود.

در این حالت سیستم عامل بیشترین وقتش را صرف این جابجایی می کند.

اصل محلی بودن:

- مراجعات به برنامه و داده ها در حافظه ،خوشه ای هستند.
- تعداد محدودی از فرآیند ها در حافظه اجرا می شوند و باید حدس زد که کدام برنامه در آینده اجرا می شود.
- به این ترتیب از کوبیدگی جلوگیری می کنند .
- تاکید در کارآمدی فرآیندها در یک محیط حافظه مجازی است.

برای کارآمدی حافظه مجازی:

- باید حمایت سخت افزاری وجود داشته باشد.
- سیستم عامل باید دارای نرم افزار هایی جهت تبادل قطعه ها از حافظه اصلی به حافظه مجازی باشد.

مشخصات صفحه بندی و قطعه بندی:

صفحه بندی ساده	قطعه بندی ساده	حافظه مجازی با قطعه بندی	حافظه مجازی با صفحه بندی
حافظه به تکه های مساوی تقسیم	حافظه اصلی تقسیم نشده	حافظه اصلی به تکه یکسان به نام قاب تقسیم شده	حافظه اصلی تقسیم نشده
برنامه به صفحه تقسیم شده	برنامه قطعه بندی شده	برنامه به صفحه تقسیم شده	برنامه قطعه بندی شده
تکه تکه شدن داخلی در قاب	بدون تکه تکه شدن داخلی	تکه تکه شدن داخلی در قاب	بدون تکه تکه شدن داخلی
بدون تکه تکه شدن خارجی	تکه تکه شدن خارجی	بدون تکه تکه شدن خارجی	تکه تکه شدن خارجی

تهیه جدول صفحه	تهیه جدول صفحه	تهیه جدول قطعه	تهیه جدول قطعه
لیست قاب های آزاد	لیست قاب های آزاد	لیست حفره های آزاد	لیست حفره های آزاد
بدست آوردن آدرس منطقی	بدست آوردن آدرس منطقی	بدست آوردن آدرس منطقی	بدست آوردن آدرس منطقی
برای اجرا باید در حافظه اصلی باشند	برای اجرا باید در حافظه اصلی باشند	برای اجرا باید در حافظه اصلی باشند	برای اجرا باید در حافظه اصلی باشند
خواندن به حافظه اصلی نیازمند دیسک است	خواندن به حافظه اصلی نیازمند دیسک است		

ساختار جدول:

- راهکار اصلی برای خواندن یک کلمه از حافظه، ترجمه یک آدرس مجازی یا منطقی به یک آدرس فیزیکی، از طریق به کارگیری جدول صفحه است.
- در حافظه اصلی قرار می گیرد.
- آدرس شروع در ثبات قرار می گیرد.
- از شماره صفحه آدرس مجازی، به عنوان شاخص برای شناسایی شماره قاب استفاده می کند.
- برای به کارگیری جداول صفحه یک سطحی یا دو سطحی

استفاده از ساختار جدول صفحه معکوس است.

- حافظه مجازی از یک حافظه پنهان مخصوص برای مدخل های جدول صفحه استفاده می شود که به آن میانگیر دم دستی ترجمه می گویند.
- یکی از راهکار های بهبود آن به کارگیری میانگیر بزرگتر با مدخلهای بیشتر است.

- یکی دیگر از راه حل ها به کار گیری اندازه صفحه بزرگ تر.
- هنگامی که صفحه در حافظه اصلی قرار نداشته باشد یک خطای دسترسی به نام خطای فقدان صفحه صادر می شود.
- یکی از تصمیمات مهم در طراحی سخت افزار تعیین اندازه صفحه است.
- اندازه صفحه به اندازه فیزیکی صفحه بستگی دارد.
- قطعه بندی این قابلیت را دارد که سیاست های اشتراکی و حفاظتی بر روی آن اجرا شوند.

طراحی بخش مدیریت حافظه:

- بستگی به سه انتخاب زیر دارد:
- اینکه از روش های حافظه مجازی استفاده شود یا خیر.
- استفاده از صفحه بندی یا قطعه بندی یا هر دو.
- الگوریتم های مورد استفاده برای جنبه های مختلف مدیریت حافظه.

سیاستهای سیستم عامل برای حافظه مجازی:

- سیاست واکشی.
- سیاست جاگذاری .
- سیاست جایگزینی.
- مدیریت مجموعه مقیم.
- سیاست پاکسازی.
- کنترل بار.

سیاست واکشی:

- صفحه های یک فرآیند می توانند به طور درخواستی وارد شوند، یا یک سیاست پیش صفحه بندی که فعالیتهای ورودی را از طریق به داخل آوردن یکباره تعدادی صفحه ها انجام می دهد.
- درخواستی.
- پیش صفحه بندی .
- اکثر سیستم عامل ها این سیاست را دنبال می کنند.

سیاست جاگذاری:

- با انتخاب یک سیستم قطعه بندی ساده، قطعه ورودی باید در یکی از فضاهای موجود در حافظه گذاشته شود.
- برای سیستم های با دسترسی غیر یکنواخت به حافظه لازم است.

سیاست جایگزینی:

زمانی که حافظه پر باشد، باید تصمیم گیری شود که کدام صفحه یا صفحاتی جایگزین شود.

الگوریتم اصلی:

- بهینه : زمان لازم برای مراجعه به آن طولانی تر باشد.
- حداقل استفاده در گذشته نزدیک. (LRU)
- خروج به ترتیب ورود. (FIFO)
- ساعت : مربوط کردن یک بیت اضافی به هر قاب .

مدیریت مجموعه مقیم:

- سیستم عامل باید تصمیم بگیرد چه مقدار از حافظه اصلی را به آن اختصاص دهد و این تخصیص می تواند به صورت پویا و یا به صورت ایستا باشد.
- اندازه مجموعه مقیم ثابت یا متغیر.
- قلمروی جایگزینی سراسری یا محلی.

مدیریت مجموعه مقیم :

جایگزینی محلی	جایگزینی سراسری	
تعداد قابهای تخصیص یافته ثابت، صفحه ای که باید تخصیص یابد از میان قابهای تخصیص یافته انتخاب می شود.	ممکن نیست.	تخصیص ثابت

تعداد قابهای تخصیص یافته می تواند متغیر باشد. صفحه ای که باید تخصیص یابد از میان قابهای تخصیص یافته انتخاب می شود.	تخصیص متغیر	صفحه ای که قرار است جایگزین شود از میان کلیه قابهای حافظه اصلی انتخاب می شود.
--	-------------	---

سیاست پاکسازی:

- بر عکس سیاست واکنشی است.
- تعیین شود که یک صفحه تغییر یافته چه موقعی باید در حافظه ثانوی نوشته شود.
- پاکسازی درخواستی: فقط زمانی یک صفحه در حافظه مجازی نوشته می شود که برای جایگزینی انتخاب شده باشد.
- پیش پاکسازی: از ابتدا قاب هایشان را در حافظه مجازی می نویسد.
- صفحه های تغییر یافته می توانند در زمان جایگزینی در خارج نوشته شوند، یا اینکه با یک سیاست پیش پاکسازی فعالیتهای خروجی از طریق نوشتن یکباره تعدادی از صفحه ها انجام گیرد.
- درخواستی.
- پیش پاکسازی.

کنترل بار:

- در مورد تعیین تعداد فرایند های مقیم در حافظه در هر لحظه است.
- درجه چند برنامه گی.

تعليق فرآیند:

اگر قرار باشد سطح چند برنامه گی کاهش یابد، یک یا چند فرایند به معلق می رود.

شش امکان جهت تعليق یک فرایند

- فرآیند با کمترین اولویت.
- فرایند خطا دهنده
- آخرین فرآیند فعال شده.

- فرایندی با کوچکترین مجموعه مقیم.
- بزرگترین فرآیند.
- فرایندی با بزرگترین پنجره اجرایی باقیمانده.

پارامترهای مدیریت حافظه:

مدخل جدول صفحه

- شماره قاب صفحه : در داخل حافظه حقیقی.
- سن: چه مدت صفحه در حافظه بوده.
- کپی در نوشتن: هنگامی که بیش از یک فرایند در یک صفحه اشتراک داشته باشند.
- تغییر: در صفحه.
- مراجعه : نشان دهنده مراجعه به صفحه.
- اعتبار : در داخل حافظه اصلی است.
- حفاظت : آیا عمل نوشتن مجاز است؟

توصیفگر بلوک دیسک

- شماره دستگاه مبادله : شماره منطقی صفحه.
- شماره بلوک دستگاه : مکان بلوک صفحه.
- نوع حافظه .

مدخل جدول داده های قاب صفحه

- حالت صفحه : آیا این قاب آزاد و... است.
- شمارش مراجع : تعداد فرآیندهایی که به صفحه مراجعه می کنند.
- دستگاه منطقی : یک کپی از صفحه.
- شماره بلوک: کپی در دستگاه منطقی .
- اشاره گر داده های قاب صفحه : به مدخلهای جدول داده ها قاب صفحه در لیست صفحه های آزاد و درهم

مدخل جدول استفاده - مبادله:

- شمارش مراجع: شماره مدخل های جدول صفحه.
- شمارش واحد صفحه / حافظه : شناسه صفحه بر روی واحد حافظه.

مدیریت حافظه در UNIX و SOLARIS:

- UNIX مستقل از ماشین می باشد ، طرح مدیریت حافظه از سیستمی به سیستم دیگر فرق می کند.
 - سیستم صفحه بندی حافظه مجازی را ارائه می کند.
- برای مدیریت تخصیص : از تخصیص دهنده حافظه هسته استفاده می کند.

سیستم صفحه بندی (از تعدادی ساختمان داده استفاده می کند).

- جدول صفحه: برای هر فرآیند .
- توصیفگر بلوک دیسک: برای صفحه فرآیند.
- جدول داده های قاب صفحه: قاب های حافظه اصلی را توصیف می کند.
- جدول استفاده - مبادله

مدیریت حافظه در windows NT:

- چگونگی تخصیص حافظه و اجرای صفحه بندی را کنترل می کند.
- هر فرآیند کاربر در NT فضای آدرس ۳۲ بیتی جداگانه را می بیند که تا ۴ گیگابایت حافظه را اجازه می دهد.
- مستندات NT این تخصیص حافظه را برای کاربران زیاد می داند.

صفحه بندی NT:

- این فضا، به صفحه هایی با اندازه ثابت تقسیم می کند.
- هر صفحه می تواند در یکی از سه حالت زیر می باشد:
- موجود: صفحه ای هنوز مصرف نشده باشد.
 - رزرو: صفحاتی که برای فرآیند رزرو شده باشد .
 - متعهد: صفحاتی که برای صفحه بندی کنار گذاشته شده باشد.
- طرح مدیریت مجموعه مقیم که استفاده شده است دارای خصوصیت تخصیص مدیریت است.

فصل نهم

زمان بندی تک پردازنده

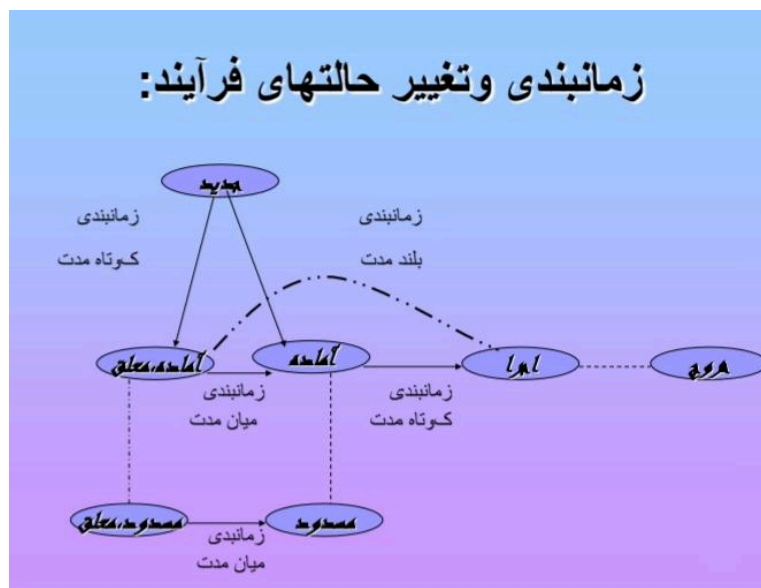
انواع زمانبندی:

- هدف از زمان بندی ،تخصیص فرآیندها به پردازنده ها در طول زمان است به گونه ای که به هدف های سیستم برسد.

انواع زمانبندی ها

- زمانبندی بلند مدت: تصمیم گیری در مورد افزودن به مجموعه فرایندها برای اجرا.
- زمانبندی میان مدت: تصمیم گیری در مورد افزودن به تعداد فرایندها.
- زمانبندی کوتاه مدت: کدام فرایند برای اجرا فرستاده شود.
- زمانبندی ورودی / خروجی: کدام درخواست ورودی و خروجی فرایند توسط کدام دستگاه اجرا شود.

زمانبندی و تغییر حالت های فرآیند:



معیارهای زمانبندی:

معیار ها از دید کاربر و مربوط به کارآمدی:

- زمان پاسخ: باید این زمان کم باشد و حداکثر سازی کاربران محاوره ای.
- زمان کل: فاصله بین پذیرش یک فرآیند تا تکمیل آن. معیاری برای کارهای دسته ای است.
- آخرین مهلت: ارضاء آخرین مهلت ها.

معیارهای دیگر از دید کاربر:

- قابلیت پیش بینی: یک کار باید در زمان یکسان و... انجام شود.

معیارها از دیدگاه سیستم و مربوط به کارآمدی:

- توان عملیاتی: تعداد فرایندهای تکمیل شده در واحد زمان.

- استفاده از پردازنده: درصد زمان مشغول بودن یک فرایند را نشان می دهد.

معیارهای دیگر از دید سیستم:

- عدالت: هر فرایند سهم عادلانه ای از منابع داشته باشد.

- اعمال اولویت ها: تقدم فرایندها با زمان بالاتر.

- توازن در منابع : باید منابع سیستم را مشغول نگهدارد.

سیاست های زمانبندی:

تابع انتخاب:

فرایند بعدی را از میان فرایندهای آماده اجرا انتخاب می کند.

سه کمیت زیر مهم است:

W = زمان سپری شده در سیستم برای انتظار و اجرا تا به حال

e = زمان سپری شده برای اجرا تا به حال

s = کل زمان مورد نیاز فرایند، که شامل e نیز هست.

حالت تصمیم گیری:

- زمانی که تابع انتخاب به اجرا در می آید را مشخص می کند . دو گروه بندی زیر را دارد:
- بدون قبضه کردن: تا تکمیل شدن ورود یا خودش برای I/O گرفتن مسدود شود.
- با قبضه کردن: تصمیم به قبضه کردن می تواند هنگام ورود و... ایجاد شود.

زمانبندی خدمت به ترتیب ورود FCFS :

- ساده ترین الگوریتم ؛

- هر فرآیند با آماده شدن به صف آماده ملحق شده وقتی فرایند جاری از اجرا باز ماند قدیمی ترین فرایند در صف در صف آماده به اجرا قرار می گیرد

زمانبندی FCFS :

$\max[w]$: تابع انتخاب

حالت تصمیم گیری : بدون قبضه

توان عملیاتی : تاکید نشده است

زمان پاسخ : می تواند زیاد باشد

سربار: حداقل

تاثیر بر روی فرآیندها: به فرایند کوتاه صدمه می زند

گرسنگی: خیر

زمانبندی نوبت گردشی RR :

- یک وقفه ساعت در فواصل زمانی دوره ای تولید می گردد
- با بروز وقفه فرایندی که در حال اجراست در صف آماده قرار می گیرد و کار آماده بعدی براساس FCFS انتخاب می شود.

زمانبندی RR :

تابع انتخاب: ثابت

حالت تصمیم گیری: با قبضه کردن

توان عملیاتی: اگر برهه زمانی خیلی کوچک باشد کم

زمان پاسخ : برای فرایندهای کوتاه زمان بسیار خوب

سربار: کم

تأثیر بر روی فرآیند ها: عملکرد عادلانه

گرسنگی: خیر

غیر انحصاری، به هر پردازش مقدار مشخصی cpu می دهند. به عبارت بهتر زمان استفاده از آن به کوانتوم هایی تقسیم می شود.

زمانبندی کوتاه ترین فرایند SPN:

- در این سیاست بدون قبضه کردن فرایند هایی که زمان پاسخ کوتاهی دارند اول اجرا می شوند. و فرآیند کوتاه از روی کارهای طولانی می گذرد و در ابتدای صف قرار میگیرد.
- امکان دارد فرایندهای بزرگ هیچگاه انجام نشود.

زمانبندی SPN:

- تابع انتخاب: $\min[s]$
- حالت تصمیم گیری: بدون قبضه کردن
- توان عملیاتی: زیاد
- زمان پاسخ: برای فرایندهای کوتاه خوب
- سربار: می تواند زیاد باشد
- تأثیر بر روی فرآیند ها: به فرایندهای طولانی صدمه زده
- گرسنگی: امکان دارد

فرایند هایی که زمان پاسخ کوتاهی دارند اول اجرا می شوند.

زمانبندی کوتاه ترین زمان باقیمانده SRT:

- فرایندی اجرا می شود که کمترین باقی مانده زمان پاسخ را داشته باشد.
- وقتی فرایندی به صف آماده وارد می شود امکان دارد زمان باقیمانده کمتری نسبت به فرایند در حال اجرا داشته باشد که با عمل قبضه کردن آن فرایند به اجرا در می آید.

زمانبندی SRT:

- تابع انتخاب: $\min[s-e]$
- حالت تصمیم گیری: با قبض کردن

- توان عملیاتی: زیاد
 - زمان پاسخ: خوب
 - سربار: می تواند زیاد باشد
 - تاثیر بر روی فرآیندها: به فرایندهای طولانی صدمه زده
 - گرسنگی: امکان دارد
- فرایندی اجرا می شود که کمتری باقی مانده زمان پاسخ را داشته باشد.

زمانبندی بالاترین نسبت پاسخ HRRN:

- اولویت دینامیک بوده و کارهای کوتاه تر اولویت بیشتر داشته و زودتر اجرا می شوند.
- فرمول تشخیص بالاترین پاسخ: $\max(w+s/s)$
- W = زمان انتظار برای پردازنده
- S = زمان خدمت مورد انتظار

زمانبندی HRRN:

- تابع انتخاب: $\max(w+s/s)$
 - حالت تصمیم گیری: بدون قبضه کردن
 - توان عملیاتی: زیاد
 - زمان پاسخ: خوب
 - سربار: می تواند زیاد باشد
 - تاثیر بر روی فرآیندها: توازن مناسب
 - گرسنگی: خیر
- اولویت دینامیک بوده و کارهای کوتاه تر اولویت بیشتر داشته و زودتر اجرا می شوند.

زمانبندی فید بک FB :

اگر هیچ نشانه ای از طول نسبی فرایندها متفاوت نداشته باشیم هیچ یک از روشهای HRRN ; SRT ; SPN را نمی توان بکار برد. راه دیگر:

از طریق جریمه کردن کارهایی که زمان طولانی تری به اجرا در آمده اند

زمانبندی FB :

- تابع انتخاب: -----
- حالت تصمیم گیری: با قبضه کردن
- توان عملیاتی: تاکید نشده است
- زمان پاسخ: تاکید نشده است
- سربار: می تواند زیاد باشد
- تاثیر بر روی فرآیندها: به نفع فرایندهای در تنگنای I/O
- گرسنگی: امکان دارد

از طریق جریمه کردن کارهایی که زمان طولانی تری برای اجرا دارند.

زمانبندی سنتی UNIX:

این سیستم برای محیط اشتراک زمانی و محاوره ای در نظر گرفته شده است.

کم اولویتها دچار گرسنگی نمی شوند.

از فیدبک چند سطحی و از زمانبندی نوبت گردشی برای هر یک از صفهای اولویت استفاده می کنند و این سیستم قبضه کردن یک ثانیه را به کار می برد.

در یک ترتیب نزولی از اولویت گروه ها به ترتیب زیر است:

-مبادله کننده.

-کنترل دستگاه I/O بلوکی.

-دستگاه پرونده.

- کنترل دستگاه I/O نویسه ای.

- فرایندهای کاربر.

موثرترین استفاده از ورودی و خروجی را فراهم می کند.

آینده خودتو امروز بساز! تخصصی ترین جزوات تجارت الکترونیک فقط در آرین فایل — اسکن کن و دانلود کن.



مشاهده جزوات تخصصی رشته تجارت الکترونیک آرین فایل