

درنسل سوم بین ۱۹۶۵ تا ۱۹۸۰ ازمدارات و جمع برای سافت کامپیوتر استفاده شد
 جاکر یکی برنامه ما به دو دستہ تقسیم می شدند. برنامه هایی که نیاز به محاسبات سنگین داشتند
 (CPU limited) و برنامه هایی که بیشتر زمان کامپیوتر را صرف ورود داده و خروج اطلاعات
 می کردند. به نسل سوم نسل چندبرنامه ای می گویند (multi programming system)

در سیستم های دسته ای ممکن بود وقت CPU به هر رود چون ممکن بود برنامه ای
 I/O limited باشد و در این هنگام CPU بیکار می ماند. برای رفع این مشکل
 ترتیب های multi programming استفاده می شد به این ترتیب حافظه به چند قسمت
 تقسیم شده و در وقت یک برنامه مجزا قرار داده می شد. بنابراین وقتی یک کاربر برای
 تکمیل عملیات ورودی خروجی I/O منتظر می ماند پردازنده به کار دیگری داده می شد.
 در سیستم های چندبرنامه ای خطای یک پردازنده وجود دارد و به کمک مکانیزم وقفه بین کارهای
 CPU limited و I/O limited تسویه می شود و بفلاهر اجرای برنامه ها
 معازی است پس بهره وری افزایش پیدا می کند.

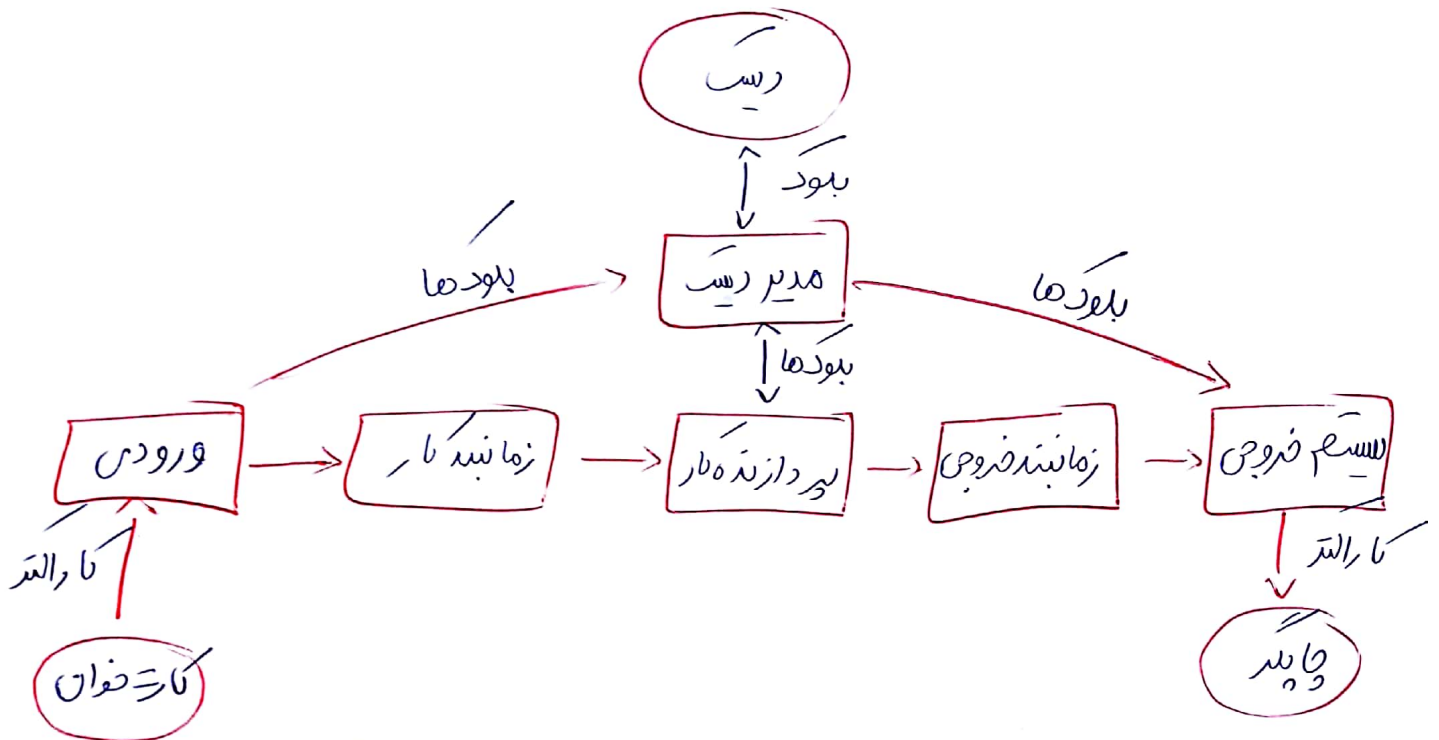
یکی از ویژگی های سیستم عامل نسل سوم آنالین اسپولینگ online spooling
 بودن است. در این سیستم بجای آنکه کارت ها از دستگاه کارت خوان وارد حافظه
 شود و مورد پردازش قرار گیرد ابتدا کارت به کارت در حافظه قرار گرفته
 و سپس بصورت بلوک در دیسک نوشته می شود. در این حالت وقتی برنامه کار بر اجرا
 می شود از سیستم عامل تقاضای ورودی می کند و اطلاعات ورودی بصورت بلوک
 و باسوی زیاد خوانده می شود.

مزایای سیستم **multi programming online spooling** نسبت به سیستم دسته ای :

- بالا بودن راندمان بالای CPU و وسایل I/O

- گردش سریع کار

- دسترسی با اولویت



سیستم ورودی : کارخانه های به توسط کارخانه فوان وارد می شود را در یکدستی جمع آوری کرده و به کمک مدیر دیسک بلودها را بر روی دیسک می نویسد و در انتها اطلاعاتی را به آن مدرک مثل محل قرارگیری آن بر روی دیسک به قسمت زمانبندی کار ارسال می شود.

زمانبندی کار : این زمانبندی یک لیست از کارهای موجود در ماشین را نگه می دارد. به این لیست **job list** می گویند. زمانبندی کار به پردازنده کار می گوید که کدام کار بعدی را اجرا کند. برای این کار اطلاعاتی در مورد محل کار و محل آن بر روی دیسک به پردازنده کار می دهد.

پردازنده کار : کار داده شده را اجراء می کند. هنگام اجرا پردازنده، فروری های خود را به صورت بلوک های بیرونی در یک سی نظمیده.

زمان بندی فروری : لیست از مدارکی که باید چاپ شوند را تهیه می دارد.

سیستم فروری : بلوک های فروری را از روی یک خواننده و کارالت به کارالت آن ها را به چاپگر می فرستد.

مدیر یک : وظیفه خواندن و نوشتن یک بلوک بیرونی در یک و تخصیص یک بلوک فانی روی یک و صی برگرداندن یک بلوک به مجموعه فضای آزاد دیگر ایدعه دارد.

سیستم زمانی (time sharing) :

در سال های ۱۹۷۰ در نسل سوم کامپیوتر ها این سیستم ها محصول شده اند. چونی در سیستم های چند برنامه ای کاربرد ارتباطی با رایانه داشت و وظایف این برنامه ها مشکلی بود (برگشت نسبتاً طولانی نتیجه کار) سیستم های اشتراک زمانی بوجود آمدند. در این سیستم ها کاربرد یک تک ترمینال، کیبورد و مانیتور با کامپیوتر ارتباط برقرار می کرد. در این سیستم ها همچنان یک CPU وجود داشت. که توسط ما اینترم های زبان بندی بیت برنامه مختلف کاربران سوئیچ می شد بنابراین کاربرد ~~تصور می کرد~~ در اختیار می است. در اینجا تا یک بیرونی میزان عملکرد کاربر است. سیستم های عامل کامپیوتر شخصی و شبکه :

نسل چهارم ۱۹۸۰ که مدارات مجتمع با مقیاس بزرگ LSI یا VLSI ابداع شد. در این نسل کامپیوتر ها از ران شد کارائی کامپیوتر ها بالا رفت محیط گرافیکی و مظهر های

۱۱۱
۱۲
مناسب روبرو گسترش بود. سیستم عامل های اولیه در این شد (Dos) فقط
تک کاربره و تک برنامه ای بودند و سیستم عامل های امروزی مانند ویندوز خاصیت های
چند کاربره و چند برنامه و شبکه را دارا بودند.

سیستم های توزیع شده : (distributed system)

سیستم توزیع شده: در سیستم های توزیع شده که در یک محیط شبکه ای اجرا می شود
 قسمت های مختلف برنامه کار بر در دین کامپیوتر مجزا اجرا می شود. کاربران از این موضوع
 بی خبرند یعنی سیستم از دید کاربر شفاف است. از مزایای سیستم توزیع شده سرعت
 بالای اجرای برنامه است. (به دلیل توزیع برنامه بین چندین کامپیوتر) این سیستم همچنین
 برای بانک های اطلاعاتی حجم مناسب است. (به دلیل توزیع بانک اطلاعات بر روی
 یک سری از کامپیوترها). دلیل استفاده از این سیستم ها:

- ۱- اشتراک منابع
- ۲- تسریع محاسبات
- ۳- قابلیت اعتماد
- ۴- ارتباطات (مثل ایمیل)

نخ thread: (multi tasking):

در تکنیک چندنخی یک فرآیند یا Process (برنامه ای که در حال اجرا است) می تواند به نخ های
 تقسیم شود که این نخ ها می توانند به صورت همزمان اجرا شوند. برنامه هایی که چند وظیفه
 مستقل از هم را انجام می دهند می توانند به صورت چندنخی نوشته شوند.

فرآیند یک برنامه در حال اجراست که منابعی از سیستم به آن تفهیم داده شده است.
 فرآیند می تواند مجموعه ای از یک یا چند نخ باشد.

لایه اطلاعات مربوط به هر پروسه یا فرآیند در جدولی به نام PCB (Process control block)
 ذخیره می شود. اطلاعاتی مانند حالت جاری پردازش، شماره شناسائی پردازش، اولویت
 پردازش، نشانی حافظه پردازش و... ذخیره می شود.

سیستم های چندپردازنده: سیستم هایی که به جای یک CPU چندین CPU داشته باشند.
 در سیستم چند پردازنده ای CPU از حافظه، امکانات ورودی و خروجی و گذرگاه های سیستم

با صورت مشترک استفاده می‌کنند.

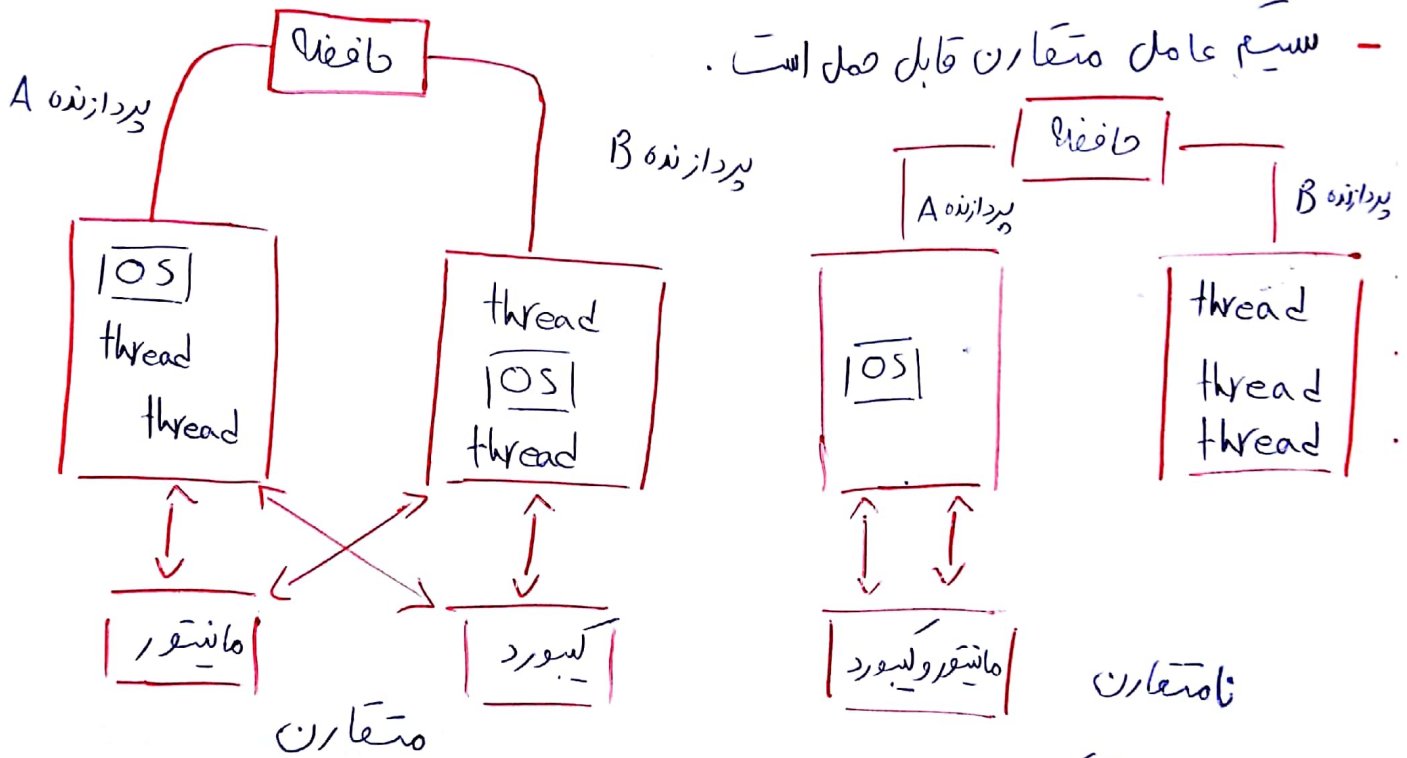
مزایای سیستم‌های چندپردازنده‌ای :

- بالا رفتن توان عملیاتی : منظور تعداد کارهایی است که در یک واحد زمانی انجام می‌شود. هرچه تعداد پردازنده‌ها بیشتر شود تعداد کارهای انجام شده در یک زمان مشخص بیشتر می‌شود. به دلیل اتلاف وقت پردازنده‌ها جهت مسائل کنترلی امنیتی و سوییچ کردن اگر تعداد پردازنده‌ها n شود سرعت اجرا n برابر نمی‌شود.
- صرفه‌جویی در هزینه‌ها : چون پردازنده‌ها از منابع تغذیه، دسیک‌ها، حافظه‌ها و... به صورت مشترک استفاده می‌کنند در هزینه‌های سخت‌افزاری صرفه‌جویی می‌شود.
- تحمل پذیری در برابر خطا : این سیستم‌ها قابلیت اعتماد را افزایش می‌دهند. در صورت خرابی یک CPU سیستم متوقف نمی‌شود و فقط ممکن است کند شود.
- سیستم‌های چندپردازنده‌ای به دو دسته متقارن **SMP** و نامتقارن **AsMP** تقسیم می‌شوند.
- سیستم‌های نامتقارن **AsMP** : یک پردازنده جهت اجرا سیستم عامل و پردازنده‌های دیگر جهت اجرای برنامه کاربردها استفاده می‌شود. چون خود سیستم عامل روی یک پردازنده اجرا می‌شود سافت این نوع سیستم عامل‌ها ساده است. یکی از مزایای سیستم عامل نامتقارن غیرقابل حمل بودن آن است یعنی برای سیستم‌های مختلف باید سیستم‌های مختلفی نوشت و سود.
- سیستم‌های متقارن **SMP** : در این سیستم‌ها سیستم عامل می‌تواند روی تمام پردازنده‌ها همزمان اجرا شود. در این حالت حافظه بین تمام آن‌ها مشترک است.
- برتری‌ها سیستم متقارن نسبت به نامتقارن :- چون سیستم عامل خود یک پردازش است پس است اگر فقط روی یک CPU اجرا شود آن پردازنده همواره بار سنگینی دارد.

در حالی که ممکن است پردازنده های دیگر بسیار باشند. بنابراین اجزای سیستم عامل روی چند پردازنده باعث معادل شدن بار سیستم می شود.

در سیستم نامتقارن اگر پردازنده اجزای لنده سیستم عامل خراب شود کل سیستم خراب می شود در حالی که سیستم متقارن اگر یک پردازنده از کار بیفتد سیستم عامل می تواند بر روی پردازنده های دیگر اجرا شود.

- سیستم عامل متقارن قابل حمل است.



سیستم های **Real time** (درنگ)

معمولاً به عنوان یک کنترل کننده در یک کاربرد خاص استفاده می شود. سیستم در این حالت می بایست در زمانی مشخص و معین جواب مورد نظر را بدهد. در صنعت موسیقی پزشکی و سیستم های کنترل صنعتی استفاده می شود. در سیستم های **Real time** معمولاً وسایل ذخیره سازی ثانویه وجود ندارد و به جایی آن های از حافظه های Ram, Rom استفاده می شود.

خدمات سیستم عامل :

- ۱- مدیریت پردازش و پردازنده : یک برنامه حاوی دستورالعمل‌هایی است که توسط CPU اجرا می‌شود. برنامه در واقع یک موجودیت Passive است. مانند محتوای یک فایل بر روی دیسک و پس پردازش یک برنامه در حال اجراست که به آن موجودیت active گفته می‌شود. یک فرآیند منابع مشخصی نیاز دارد. مثل CPU، حافظه، و فایل‌ها.
- پردازش واحد کاربر سیستم است. سیستم عامل در رابطه با مدیریت پردازش وظایف زیر را بر عهده دارد:

- ایجاد و حذف پردازش‌های کاربر و سیستم

- زمان بندی پردازش‌ها

- مدیریت هم‌زمانی پردازش‌ها و ارتباط بین آن‌ها (جلوگیری از بن بست)

dead lock

مفهوم بن بست : حالتی است که پردازش‌ها در یک طقه بسته منتظر تمام شدن وظایف هم‌دیده هستند. بنابراین هیچ‌یک نمی‌توانند اجرا شوند و سیستم قفل می‌شود.

۲- مدیریت حافظه : CPU به‌طور مستقیم فقط با حافظه اصلی سروکار دارند و برنامه‌ها

بهت اجرا در حافظه اصلی قرار می‌گیرند. سیستم عامل باید مشخص کند هر بخش از حافظه توسط

چه پردازشی استفاده می‌شود. تخصیص و باز پس‌گیری فضای حافظه و محافظت از آن داخل

برآیندها از وظایف این بخش از سیستم عامل است.

۳- مدیریت فایل : جهت استفاده ساده از اطلاعات کامپیوتر سیستم عامل زیر منطقی

یکسانی از اطلاعات ذخیره شده روی انواع وسایل ذخیره سازی مثل هارد، دیسک نوری و...

پدید می‌آورد. در واقع سیستم عامل فراموش فیزیکی وسایل را از دید کاربر مخفی می‌کند و یک

واحد ذخیره منطقی به نام فایل ارائه می‌دهد. سیستم عامل در رابطه با مدیریت فایل وظایف

زیر را انجام می‌دهد :

۲ ایجاد و حذف فایل - ایجاد و حذف دایرکتوری - انجام عملیات کپی و انتقال
تغییرات بر روی فایل و دایرکتوری - مدیریت قرارگیری فایل‌ها بر روی رسانه‌ها - مدیریت
دسترسی‌های مختلف به فایل‌ها

۴- مدیریت وسایل ورودی خروجی I/O :

هر دستگاه ورودی یا خروجی برای کار کردن به مجموعه‌ای دستورالعمل و یا کنترل‌های خاص
نیاز دارد که سیستم عامل وظیفه برداشتن به این جزئیات و کنترل‌های خاص را برعهده
می‌گیرد. در نتیجه برنامه‌نویس و کاربر نیاز به دانستن ارتباطات سطح پایین وسایل ورودی
خروجی را ندارد.

وظایف فرعی سیستم عامل : تشخیص خطاهایی مثل می‌تواند در CPU و حافظه اتفاق
بیفتد مثل خراب شدن بیت از حافظه، خطاهایی در وسایل I/O مثلاً نویزی که
بر روی کابل پرینتر افتاده باشد یا چاپگر کاغذ نداشته باشد.

فرمان : یکی از مهم‌ترین برنامه‌های سیستم عامل مفسر فرمان است که در واقع
واسطه بین کاربر و سیستم عامل می‌باشد. مفسر دستورات کاربر را گرفته و اجرا می‌کند.
فرمان‌ها از طریق صف‌های ورودی و خروجی متن‌ی بر روی صف‌های نمایش داده می‌شود.
تشخیص اتمام انتقال داده‌ها : به روش (Polling) یا سرکشی و (Interrupt) وقفه انجام
می‌شود.

روش Polling : در این روش هرگاه دارای بیانی کنترلی است که CPU با بررسی
آن می‌تواند متوجه شود که آیا عملیات انتقال تمام شده یا خیر. در نتیجه در این تکنیک
CPU مرتباً بیت‌های کنترلی دستگاه‌های مختلف را بصورت سرکشی بررسی می‌کند.
سرعت آن پایین می‌باشد.

۳- **interrupt** : در این تکنیک هر دستگاه دارای سیگنال کنترلی مخصوص به خود است که به نوبت با CPU ارتباط دارد. هرگاه انتقال داده توسط آن دستگاه تمام شود سیگنالی را به نام وقفه به سمت CPU می فرستد و CPU را مطلع می سازد.

وقفه **interrupt** : وقفه ها جزو مهمی از معماری کامپیوتر هستند. نحوه عملکرد آن ها از ماشین به ماشین دیگر متفاوت است. وقفه را یکبار می سازد تا اجزای دستور العمل های جاری پردازنده موقتاً متوقف شود و دستورات سرویس دهی دیگر اجرا گردد و پس از آن کنترل دوباره به برنامه وقفه داده شده بازگردد.

انواع وقفه ها :

۱- وقفه برنامه (**Program check**) : به دلیل اجرای بعضی دستورات مخفی در سیستم

سرریز شدن محاسباتی و اجرای دستور العمل غیر مجاز و تقسیم بر ۰ و رجوع به آدرس خارج از محدوده مجز کاربر (به این وقفه ها اغلب trap یا تله می گویند)

۲- وقفه زمان سنج (**timer**) : این وقفه به سیستم عامل امکان می دهد بعضی اعمال را مرتباً انجام دهد.

۳- وقفه های I/O : این وقفه ها به صورت کنترل کننده های دستگاه های I/O تولید می شوند تا کامل شدن طبعی یک عمل یا خطا در عمل را نشان بدهند.

۴- وقفه نقص سخت افزار : وقفه هایی که در اثر نقص هایی مانند نقص برق

اتفاق می افتد

۵- وقفه **SVC** : یک تقاضا از طرف برنامه کاربر جهت دریافت سرویس ویژه ای از سیستم عامل می باشد.

۶- وقفه **Restart** : با فشار دادن دکمه ریست اجرا می شود.

۴
در تقسیم بندی کلی می توان وقفه ها را به سه دسته تقسیم کرد:

(الف) وقفه های داخلی trap: برای اجرای خود برنامه بصورت داخلی در CPU رخ می دهد.

(ب) وقفه های خارجی: از دستگاه های خارجی، خارجی و تایمر و صفحه کلید و صفحه های سخت افزار می ناسی می شود.

(ج) وقفه های نرم افزار SVC: برای فراخوانی توابع سیستمی توسط برنامه رخ می دهد.

مراد عملیات I/O به نکت وقفه ها:

- ۱- CPU دستگاه جانبی را مقداردهی می کند و تقاضای شروع عمل I/O را به دستگاه می فرستد.
- ۲- پردازنده به سران پردازش دیتای می رود و دستگاه جانبی نیز با خود را انجام می دهد.
- ۳- دستگاه I/O با اتمام کار با فرستادن سیگنال وقفه این موضوع را اطلاع می دهد.
- ۴- پردازنده در اولین فرصت به وقفه صادر شده سرویس می دهد.

کانتکس سوئیچ (context switch):

قبل از آنکه سیستم عمل کنترل را به روال وقفه ای بفرستد وضعیت پردازش جاری را در محلی صفای می کند تا بعداً بتواند آن را ادامه بدهد. به این عملیات تعویض متن می گویند. برای عملیات تعویض متن از بیت **Program status word (PSW)** استفاده می شود که ترتیب اجرای دستورات را کنترل می کند.

انواع سیستم عامل ها از نظر ساختاری:

- ۱- یکپارچه (monolithic)
- ۲- لایه ای (layered)
- ۳- ماشین های مجازی (Virtual machine)
- ۴- کلیت سروری (Client/server)

۱- یکپارچه: سیستم عامل بصورت یک مجموعه از رول ها نوشته شده که هر یک از آن رول ها

می تواند دیتای را به هنگام نیاز فراخوانی کند. برای مخفی کردن اطلاعات امکاناتی وجود ندارد و هر رول برای دیگر رول ها قابل مشاهده است.

۲- لایه ای: در روش لایه ای سیستم عامل به تعدادی سطح یا لایه تقسیم می شود که هر کدام در بالای لایه پایین تر قرار می گیرند. مزیت مهم این روش پیمانه ای بودن

۲- **(modularity)** بودن است یعنی لایه ها به گونه ای تقسیم می شوند که هر لایه

فقطا توابع و سرویس های لایه پایین تر را استفاده می کند. به این ترتیب هر لایه را می توان مستقل از لایه های دیگر طراحی کرد.

۳- ماشین مجازی: هسته سیستم که به ماشین مجازی معروف است بدروسی سفت افزار اجرا شده و چند برنامه را پدید می آورد. این ماشین مجازی برای کاربران مشابه یک نسخه از سیستم است.

۴- **کلاش سرور**: روند طراحی سیستم عامل ها همواره با این ایده همراه بوده تا جایی که ممکن است که با لایه های بالاتر منتقل شوند تا یک هسته پدید آید. برای این منظور اکثر وظایف سیستم عامل را در سطح کاربر و مشابه پردازش های کاربران پیاده سازی می کنند.

فصل ۳

زمانبندکار **(job scheduler)**:

زمانبندکار رویینی است که بر اساس الگوریتمی خاص یکی از کارهای موجود در جدول را انتخاب کرده و جهت اجرا آنرا به حافظه می آورد تا بعداً تبدیل به پردازش شود. کارهایی که منتظر اجرا شدن هستند اصطلاحاً **Hold** هستند و هنگامی که جهت اجرا انتخاب می شود اصطلاحاً به حالت **Ready** می روند. الگوریتم های زمانبندی کار یا روش های انتخاب طاهز:

۱- روش **FIFO**: در این الگوریتم اولین کاری که وارد می شود انتخاب شده و تبدیل به پردازش می شود.

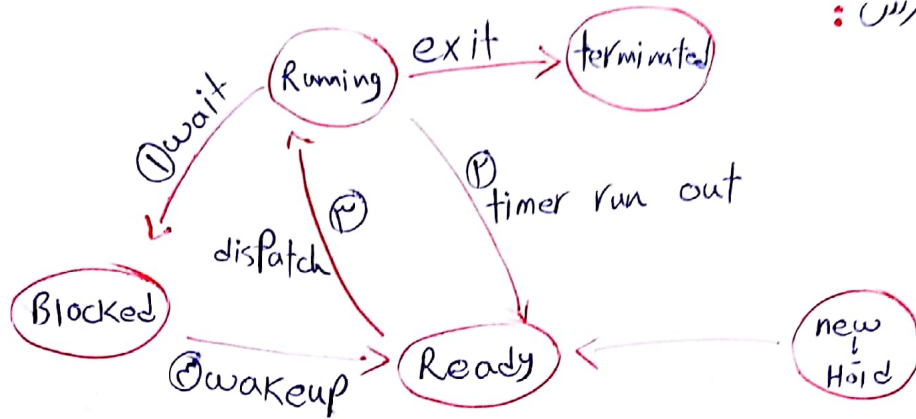
۲- روش **STF**: در این الگوریتم کوتاه ترین کاری که وجود دارد پیداشده و جهت اجرا

۳

انتخاب می شود

۳- روش ترلیب: در این روش ترلیب از کارهای **CPU limited** و **I/O limited** جهت اجرا انتخاب می شود.

حالات یک پردازش:



یک پروسس در زمان حیات خود می تواند در یکی از سه وضعیت (اجرا، آماده، بسته) قرار بگیرد. هنگامی که دستور اجرایی پردازشی صادر می شود و یا زمان بندی کاری را انتخاب می کند جهت اجرا، این پردازش به صف آماده وارد می شود و منتظر CPU می ماند. در انتقال (سه-۲) به پروسس داده می شود تا اجرا گردد. انتقال (دو-۲) زمانی صورت می گیرد که بیش از زمان پردازش تمام شده و CPU باید از آن گرفته شود لذا پردازش از حالت Running به حالت Ready می رود. در حالت Ready پردازش تمام منابع مورد نیاز را به جز CPU در اختیار دارد. هنگامی که لازم باشد پردازش در حال اجرا برای بروز رخدادی (مثل تکمیل انتقال اطلاعات) صبر کند آن ناه از طریق انتقال (یک-۱) از حالت Running به حالت Blocked می رود. پس از رفع علت انتقال، (مثلاً وقفه ای اعلام می کند انتقال پایان یافته) پردازش از طریق انتقال (چهار-۴) به حالت آماده رفته و دوباره در صف انتقال CPU قرار می گیرد.

FIFO

	P_1	P_2	P_3	P_4
time	20	2	3	2

مثال: متوسط زمان انتظار برای این چهار پروسه را حساب کنید

جواب: $\rightarrow$

P_1	P_2	P_3	P_4
20	22	25	27

$$\frac{0 + 20 + 22 + 25}{4} = 16.75$$

متوسط زمان انتظار

زمان بندی Round Robin (چرخشی) یا RR

این زمان بندی غیر انحصاری است در این زمان بندی با تخصیص که انقضای زمانی یا بیش زمانی

(Time slice) وقت پردازنده به صورت عادلانه بین پردازش تقسیم می شود.

مثال: پردازش های به صورت زیر وارد می شوند با استفاده از زمان بندی RR

و بارش زمانی 4 میلی ثانیه متوسط زمان انتظار را بدست آورید.

	P_1	P_2	P_3	P_4
Time	20	2	3	2

P_1	P_2	P_3	P_4	P_1	P_1	P_1	P_1
4	6	9	11	15	18	23	27

نسبت پروسه زمانی به زمان تقویم متن بستگی بسیار زیاد باشد تا زمان

$$\frac{0 + (11 - 4) + 4 + 6 + 9}{4} = 6.5$$

پردازنده ماشین پیدا کنند

متوسط زمان انتظار

PAPCO

* (1)

$$\frac{27 + 6 + 9 + 11}{4} = 13.25 \text{ ms}$$

متوسط زمان بازگشت
بردارش می‌فوق را

مثال با استفاده از راندر این (RR)
و که انقوع زمان ۴ میلی ثانیه

	P ₁	P ₂	P ₃
	24	3	3

* متوسط زمان انتظار

* متوسط زمان بازگشت

$$\frac{24 + 3 + 3}{3} = 10$$

* متوسط زمان اجرا

P ₁	P ₂	P ₃	P ₁	P ₁	P ₁	P ₁	P ₁
4	7	10	14	18	22	26	30

$$\frac{30 + 7 + 10}{3} = 15.66$$

متوسط زمان بازگشت

$$\frac{0 + (10 - 4) + 4 + 7}{3} = 5.66$$

متوسط زمان انتظار

$$\text{متوسط زمان بازگشت} = \text{متوسط زمان اجرا} + \text{متوسط زمان انتظار}$$

مثال: روش (RR) با که انقوع ۲ms

	P ₁	P ₂	P ₃	P ₄
Time	6	3	1	7

$$\frac{7 + 1 + 6 + 3}{4} = 4.25$$

متوسط زمان اجرا
* متوسط زمان بازگشت
* متوسط زمان انتظار

P_1	P_2	P_3	P_4	P_1	P_2	P_4	P_1	P_4	P_4
2	4	5	7	9	10	12	14	16	17

$$17 + 14 + 10 + 5 = 11.5$$

متوسط زمان بازگشت

$$\frac{4}{P_1} [0 + (7-2) + (12-9)] + \frac{2}{P_2} [2 + (9-4)] + \frac{P_3}{4} [5 + (10-7)] + \frac{12-14}{P_4}$$

$$(14-12)$$

$$7.25$$

زمان انتظار

SPN

زمان بندی S.J.F (shortest job first)

این زمان بندی انحصاری است در این روش پردازنده به پردازش های با زمان

کمتر تخصیص میابد

نکته: مشکل این روش این است که با اولویت پردازش های که فیکته فرایند می طولانی تر

دچار starvation (مقطعی زدگی) می شود

	P_1	P_2	P_3	P_4	SJF
time	20	2	3	2	پردازش های زیر و مورد دارند، باروش میانگین زمان انتظار و بازگشت را حساب میکنند

$$\frac{P_2}{2} + \frac{P_4}{4} + \frac{P_3}{7} + \frac{P_1}{27} = 3.25$$

$$\frac{27 + 2 + 7 + 4}{4} = 10$$

مثال: زمان انتظار را با استفاده از SJF به دست آورید

	P ₁	P ₂	P ₃	P ₄
time	6	8	7	3

P ₄	P ₁	P ₃	P ₂
0	3	9	16
			24

$$AVG(W) = \frac{3 + 16 + 9}{4} = 7$$

FILO	P ₁	P ₂	P ₃	P ₄
0	6	14	21	24

$$AVG(W) = \frac{0 + 6 + 14 + 21}{4} = 10.25$$

تکنیک aging: به منظور جلوگیری از قحطی از تکنیک aging استفاده می شود در این تکنیک به تدریج اولویت پروسس هایی که مدت زیادی منتظر بوده اند

افزایش داده می شود مثلاً هر n دقیقه یکبار یک واحد به اولویت پروسس

های منتظر در صف Ready افزوده می شود.

زمان بندی SRT (shortest Remaining time):

این همان بندی مختصرانی است در آن از صف آماده فرایندی انتخاب

می شود که کمترین زمان باقی مانده برای فاکتوری ابرار داشته باشد

Subject

Date

در این حالت سایر فرآیندهای جدید الوقوع به صف آماده نیز در نظر گرفته می شود
(اولویت فرآیندها فرآیندهای است که کمتر به زمان باقی مانده را داشته باشند).



مرداد

پنجشنبه

ALVAN
Paint & Resin Manufacturer1395 / 5 / 18
2016 / 7 / 29
1427 / 10 / 24

→ جواب

P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_1	P_3	P_4
2	4	5	7	9	10	12	14	16	17

$$\frac{[5 + (7-2) + (12-9)] + [2(9-4)] + 4 + [5 + (10-7) + (14-12)]}{4}$$

= 7.25 → زمان انتظار

$$\frac{14 + 10 + 5 + 17}{4} = 7.5$$

زمانبندی JF (shortest job first)

این زمانبندی انحصاری است. در این روش پردازنده به پردازش‌هایی با زمان کمتر تخصیص می‌یابد. مشکل این روش این است که با اولویت پردازش‌های کوچکتر فرآیندهای طولانی‌تر دچار starvation یا قحطی زدگی می‌شوند.

جمعه



مرداد

1395 / 5 / 18

2016 / 7 / 29

1427 / 10 / 24

مثال 4: پردازش‌های زیر موجود دارند با روش گجی می‌توان زمان

انتظار و بازگشت را می‌سبب کنید

P_1	P_2	P_3	P_4
20	2	3	2

21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---

شهادت حضرت امام جعفر صادق علیه السلام (۱۴۸ هـ ق) (تعطیل) - روز اهدای دم

P_2	P_4	P_7	P_1
۰	۲	۴	۷
۰	۲	۴	۷

جواب ←

$$\frac{0 + 2 + 4 + 7}{4} = 3.25 \quad \text{انتظار}$$

$$\frac{27 + 2 + 7 + 4}{4} = 10 \quad \text{بزرگست}$$

سوال: زمان انتظار را با استفاده از گزینش به دست آورید

P_1	P_2	P_4	P_7
۶	۸	۷	۳

$$\frac{3 + 9 + 16 + 0}{4} = 7 \quad \text{زمان انتظار}$$

تکنیک aging: به منظور جلوگیری از تسلی از این تکنیک استفاده می شود

در این تکنیک به قدری اولویت پردازش هایی که مدت حساب منتظر پردازش در آن درازتر باشد
افزایش داده می شود مثلاً عدد ۱۱ دقیقاً یکبار یک واحد به اولویت حسابی
پردازش حسابی منتظر در صف و ready انجام می گیرد

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶
۲۲	۲۳	۲۴	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰	۳۱	۱	۲	۳	۴	۵	۶

زمان بندی SRT (Shortest Remaining Time)

این زمان بندی غیر انحصاری است در آن از صف آماده فرآیندی انتخاب می شود که کمترین زمان باقی مانده برای خاتمه اجرا داشته باشد. در این حالت به فرآیندهای جدید و الوقوع به صف آماده نیز در نظر گرفته می شوند. (اولویت فرآیندهای فرآیندی است که کمترین زمان باقی مانده را داشته باشد)

مثال: به وسیله زمان بندی SRT میانگین زمان انتظار را حساب کنید

	P_1	P_2	P_3	P_4
Time	8	4	9	5
زمان ورود	0	1	2	3

→ جواب

P_1	P_2	P_4	P_1	P_3
0	1	0	10	17
				24

زمان ورود: P_1 زمان ورود: P_2 زمان ورود: P_3 زمان ورود: P_4

$$[0 + (10 - 1) - 0] + [(1 - 1)] + [(17 - 2)] + [(5 - 3)] = 6.5$$

زمان انتظار: $\frac{6.5}{4}$

حل به وسیله گزینش :

P_1	P_2	P_4	P_3
11	12	17	26

زمان بندی با اولویت بالاترین نرخ پاسخ

زمان بندی (HRRN) highest response Rating Next :

این زمان بندی از آن جهت است و این زمان بندی بر مبنای ارزشهای الگوریتم گزینش را تا حدودی حل می کنند مانند Round Robin. اولویت در این زمان بندی داینامیک بوده و از عوامل زیر محاسبه می شود :

فرموله

$$\text{اولویت اجرایی پردازش} = \frac{\text{زمان سرویس} + \text{زمان انتظار}}{\text{زمان سرویس}}$$

نکته: چون زمان اجرا در مخرج است لذا کارهایی با زمان کوتاه تر اولویت بیشتری پیدا می کنند.

نکته: چون زمان انتظار در صورت گسرهاست لذا کارهایی با زمان طولانی تر نیز در بازه زمانی مختلف اولویت اجرا پیدا می کنند

یادداشت

زمان بندی اولویت (priority schedule): در این زمان بندی اجرای فرآیندها با توجه به

اولویت آنها انجام می شود. اگر به صورت انحصاری باشد پردازش با اولویت

بیشتر جلوی صف ready قرار می گیرد و اگر به صورت غیر انحصاری باشد

پردازشی که وارد صف ready می شود اگر اولویت بالاتری داشته باشد پردازش را

در اختیار می گیرد. این زمان بندی اغلب به صورت انحصاری پیاده سازی می شود

مثال: میانگین زمان انتظار را برای پردازش های زیر در سیستم اولویت بندی به دست آورید

	P_1	P_2	P_3	P_4	P_5
Time	10	1	2	1	5
مقدار اولویت	3	1	3	4	2

(انحصاری)

اگر اولویت ها یکسان بود

به صورت FIFO عمل می کنیم

→ جواب

P_2	P_5	P_1	P_3	P_4
0	1	6	16	18
				19

$$\text{زمان انتظار} = \frac{6 + 10 + 16 + 18 + 1}{5} = 8.2$$

زمان پردازش (Longest processing Time) LPT ؟ ۱. تخصیص

در این زمان بندی پردازش هر یک از زمان پردازش تا اولویت اجرا دارند و این زمان بندی غیر بهینه است و می تواند برای سیستم های با پردازش متعادل یکپارچه

$$\text{زمان پردازش} = \frac{\text{زمان کل}}{\text{CPU}}$$

$$\text{توان عملیاتی} = \frac{\text{تعداد پردازش های سیستم}}{\text{زمان کل}}$$

$$\text{Optimized Response Time} = n * \text{Time slice}$$

↓ تعداد پردازش های موجود در سیستم

۱. تخصیص

زمان بندی **Lottery** در این زمان بندی به هر پردازش اعدادی نسبت داده می شود

تعداد اعداد نسبت داده شده به هر پردازش وابسته به اولویت آن می باشد

یعنی هر قدر اولویت پردازش بیشتر باشد اعداد بیشتری به پردازش داده می شود

سیستم به طور اتفاقی یک عدد تولید کرده و پردازش که آن عدد را دریافت دارد

CPU را دریافت دارد

یادداشت

هر قدر اولویت برنامه بیشتر باشد، شانس آن برای دریافت داشتن

عدد اتفاقی تولید شده عدد نتیجه دریافت گرفتن پردازنده بالاتر می رود

مرداد

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰	۳۱	۳۲	۳۳	۳۴	۳۵	۳۶	۳۷	۳۸	۳۹	۴۰	۴۱	۴۲	۴۳	۴۴	۴۵	۴۶	۴۷	۴۸	۴۹	۵۰	۵۱	۵۲	۵۳	۵۴	۵۵	۵۶	۵۷	۵۸	۵۹	۶۰	۶۱	۶۲	۶۳	۶۴	۶۵	۶۶	۶۷	۶۸	۶۹	۷۰	۷۱	۷۲	۷۳	۷۴	۷۵	۷۶	۷۷	۷۸	۷۹	۸۰	۸۱	۸۲	۸۳	۸۴	۸۵	۸۶	۸۷	۸۸	۸۹	۹۰	۹۱	۹۲	۹۳	۹۴	۹۵	۹۶	۹۷	۹۸	۹۹	۱۰۰
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

زمانه صف چندگانه (Multiple Queues) : در این الگوریتم صف آماده به صف های جداگانه مختلفی تجزیه می شود و هر پردازش وارد یک صف می گردد. هر صف زمانبندی خود را دارد

زمانه صف چندگانه با فیدبک Queue Multiplex level feedback : در روش قبل هنگامی که یک پردازش وارد صفی می شود نمی تواند از آن صف خارج شود ولی در زمانبندی MFA پردازش ها می توانند بین صف ها جا به جا شوند

تجزیه صف به ریز صف ها : در بنیست یکی unit 4 ⇒ dead block
 مفهوم dead block : اگر هر پردازش متعلق به یک مجموعه به انتظار حادثه ای است که تنها توسط پردازش دیگری در آن مجموعه می تواند رخ دهد حالت بنیست پدید می آید

شرایط لازم برای ایجاد dead block : شرایط کافی
 ! انحصار متقابل : یعنی حداقل یک منبع باید مانع قابل اشتراک باشد یعنی اگر منبع در اختیار پردازشی بود و پردازش دیگری درخواست آن را کرد
 باید درخواست کننده به تاخیر بیفتد تا منبع آزاد شود به عبارت دیگر نقطه یک پردازش در هر زمان می تواند از منبع استفاده کند
 این عامل بخصوص بالقوه در بروز بنیست است

جمعه ۱۵ مرداد

۱۳۹۵ / ۵ / ۱۵
 2016 / 8 / 5
 ۱۴۳۷ / ۱۱ / ۲

۱۸	۱۷	۱۶	۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱
۳۱	۳۰	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۱۴

۱. **آگرسشن و منتظر ماندن** : یعنی فرآیندی منبغی را در اختیار دامنه باشد و منابع دیگر را در فرآست کند. این نیز بالقوه است

۲. **اشخاصی بودن** : منبع به صورت داوطلبانه و سب از خانه کار فرآیند بازگردانده خواهد شد و منبع را نمی توان به اجبار از پروژس گرفت ← بالقوه

۳. **انتظار می رفتی** : باید مجموعه ای از پروژس های $\{p_0, p_1, \dots, p_n\}$ وجود داشته باشد به طوری که p_0 منتظر منبغی از p_1 و p_1 منتظر منبغی از p_2 و ... و p_n منتظر منبغی از p_0 است ← بالفعل است

۴. **منبع بحرانی** : (critical Resource) همان منبغی است که به سبب آن رعایت وجود دارد

۵. **بخش بحرانی (critical Area)** : بخش از فرآیند است که با منبع بحرانی در ارتباط باشد

۶. **گرسنگی (starvation)** : منبغی پیش از حد متعارف به فرآیندی تخصیص داده نشده و غذا پیدا کند برای گرفتن آن در انتظار بماند

۷. **گراف تخصیص منابع (resource allocation graph)** : در این گراف گره ها از پروژس ها و منابع تشکیل می شوند. پروژس ها به صورت دایره و منابع به شکل مستطیل ترسیم می شوند. مقدار سوزن های که منبع را به صورت نقطه درون مستطیل ها نشان می دهند

۱. اگر منبع و منتقلی منابع: یعنی فرآیندی منبعی را در اختیار داشته باشد و منابع دیگر

در فرآیند است. این نیز بالقوه است

۲. انحصاری بودن: منبع به صورت داوطلبانه و وسیله از خانه کار فرآیند بازگرداندن خواسته

و منبع را نمی توان به اجهار از پروژه ای گرفت ← بالقوه

۳. انتظار در منبع: باید مجموعه ای از پیردازش های $\{p_0, p_1, \dots, p_n\}$ وجود داشته باشد

به طوری که p_0 منتقلی منبعی از p_1 و p_1 منتقلی منبعی از p_2 و ... و p_n منتقلی منبعی از p_0 است

← بالفعل است

منبع بحرانی: (critical Resource): همان منبعی است که به سبب آن

رعا بست وجود دارد

بخش بحرانی (critical Area): بخش از فرآیند است که با منبع بحرانی در ارتباط

باشد

گرسنگی (starvation): منبعی پیش از حد متعارف به فرآیندی تخصیص داده نشده

و غذا آیند دیگر برای گرسنگی آن در انتظار بماند

گراف تخصیص منابع: (resource allocation graph): در این گراف گره ها از پروژه ای

یادداشت

ها و منابع تشکیل می شوند. پیردازش ها به صورت دایره و منابع به شکل مستطیل ترسیم

می شوند. مقدار مؤنه های یک منبع را به صورت نقطه درون مستطیل ها نشان می دهند.

۱ گذشتن و منتظر ماندن : یعنی فرد آیندی منبعی را در اختیار داشته باشد و منابع دیگر را در اختیار نداشته باشد. این نیز بالقوه است

۳ انحصاری بودن : منبع به صورت داوطلبانه و سب از خانه کار فرم آید بازگردانده خواهد شد و منبع را نمی توان به اجبار از پروژه ای گرفت ← بالقوه

۴ انتظار می رفتی : با یک مجموعه ای از پیدازس های $\{p_0, p_1, \dots, p_n\}$ وجود داشته باشد به طوری که p_0 منتظر منبعی از p_1 و p_1 منتظر منبعی از p_2 و ... و p_n منتظر منبعی از p_0 است ← بالفعل است

منبع بحرانی (critical Resource) : همان منبعی است که بدسترس نباشد آن وقت باعث وجود دارد

بخش بحرانی (critical Area) : بخشی از فرایند است که با منبع بحرانی در ارتباط باشد

گذشتگی (starvation) : منبعی پیش از حد متعارف به فرآیندی تخصیص داده نشود و غذا نیندیکه برای گذشتن آن در انتظار بماند

گراف تخصیص منابع (resource allocation graph) : در این گراف گره ها از پیدازس ها و منابع تشکیل می شوند. پیدازس ها به صورت دایره و منابع به شکل مستطیل ترسیم می شوند مقدار مؤنه های یک منبع را به صورت نقطه درون مستطیل نشان می دهند

۱۷	۲۲	۱	۱۸	۲۳	۲	۱۹	۲۴	۳	۲۰	۲۵	۴	۲۱	۲۶	۵	۲۲	۲۷	۶	۲۳	۲۸	۷	۲۹	۸	۳۰	۹	۳۱	۱۰	۱	۱۱	۱۲	۲	۱۳	۳	۱۴	۴	۱۵	۵
----	----	---	----	----	---	----	----	---	----	----	---	----	----	---	----	----	---	----	----	---	----	---	----	---	----	----	---	----	----	---	----	---	----	---	----	---

هنگامی که پردازش P_i نمونه ای از منبع خروجی R_j را دریافت می کند یک گمان جهت دار از پردازش P_i به مستقل منبع R_j وصل می کند و هنگامی که نمونه ای از منبع R_j به پردازش P_i تخصیص میابد یک گمان جهت دار از نقطه ای درون مستقل R_j به محیط دایره P_i ترسیم می گردد

$$P_i \rightarrow R_j$$

$$R_j \rightarrow P_i$$

پردازش به دایره
منابع به مستقل

مثال) گراف تخصیص منابع را بر این سیستم زیر رسم کنید

$$P = \{P_1, P_2, P_3\}$$

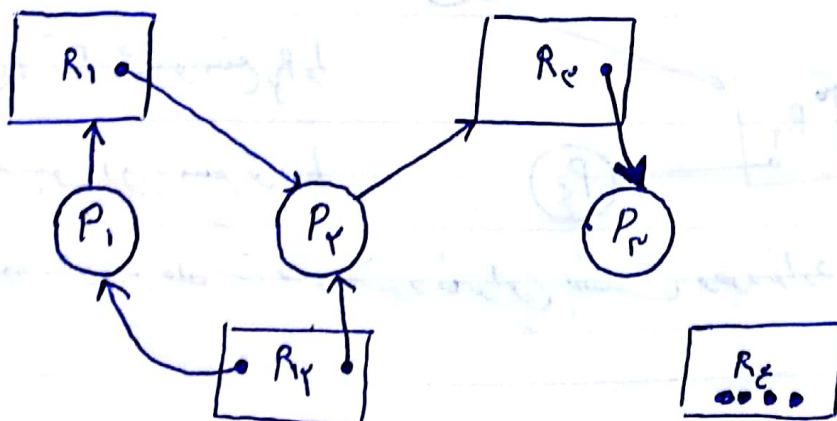
پردازشها

$$R = \{R_1 \text{ (یک نمونه)}, R_2 \text{ (دو نمونه)}, R_3 \text{ (سه نمونه)}, R_4 \text{ (یک نمونه)}\}$$

$$E = \{P_1 \rightarrow R_1, P_2 \rightarrow R_2, R_1 \rightarrow P_2, R_2 \rightarrow P_2, R_2 \rightarrow P_1, R_3 \rightarrow P_3\}$$

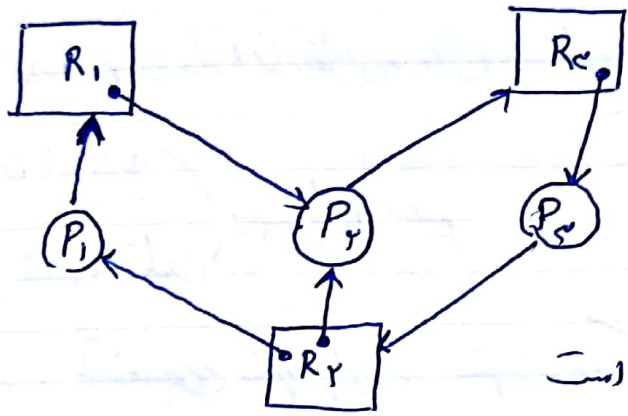
$$R_4 \rightarrow P_3$$

=> جواب



نکته: اگر در گراف حلقه وجود نداشته باشد سیستم system در حالت بنبست نیست

مساله



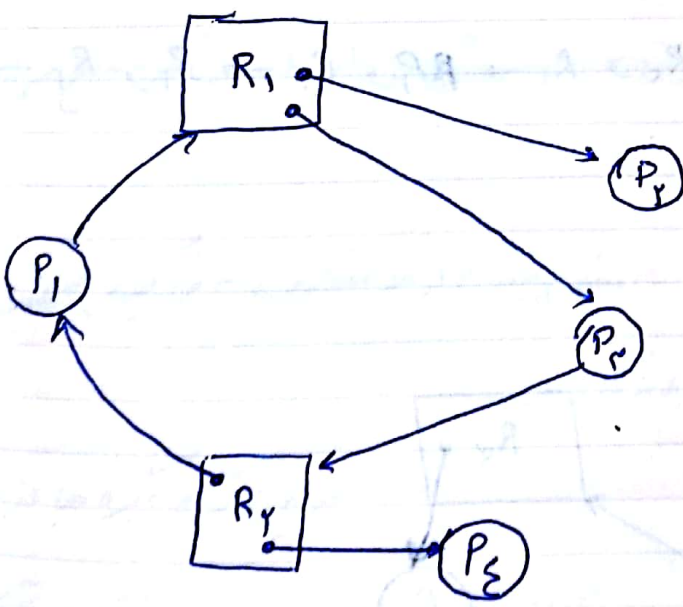
هر سه P و $deadblock$ هستند

چرا که به دازس P_2 منتظر

R_2 است که خود در اقصیه P_2 است

از طرف دیگر P_2 خود منتظر P_1 یا P_3 است

که منبع R_2 را آزاد کند همچنین P_1 منتظر P_3 است تا R_1 را رها کند



با اینکه در این شکل حلقه وجود

دارد ولی از آنجا که ممکن است

به دازس P_4 تمام شده و منبع R_2 را

آزاد کند و سپس آن منبع توسط

P_2 استفاده شود و حلقه شکسته شود بنابراین بنبست وجود ندارد

۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱
۵	۴	۳	۲	۱	۳۱	۳۰	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲
۲	۱	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷

- ۱- حرف نقر زین ب
- ۲- حلو بکر ازین ب
- ۳- اجاب ازین ب
- ۴- کف با انا بکرین ب

- * ۱- دارن اثر در معنی (Technical) بر ستاد با یغین ب استفاده دهنده دست و معنی ب
- * ۲- دارن اثر در معنی (Reset) دست
- * ۳- دارن اثر در معنی (Reset) دست
- * ۴- دارن اثر در معنی (Reset) دست

* ۵- دارن اثر در معنی (Reset) دست

- ۱- حلو بکر از انا بکرین
- ۲- Hold & wait
- ۳- حلو بکر از انا بکرین
- ۴- حلو بکر از انا بکرین

* 1 - در این مورد هر دو منابع احادیث یک مضمون بصورت انفرادی استناد می‌دهند

مانند استناد انفرادی از منابع Read-only

* 2 - در این موردی خود سند العمل و در سند العمل لول فرزند ترناد و سند ترناد

که تمام منابع (منابع مورد نظر) آزاد باشند

در سند العمل عدم تبعیت از سند عملی در این مورد است که اگر از منابع انفرادی است
است در سند عملی تبعیت از سند عملی است با این منبع عدم تبعیت از سند عملی

* 3 - در این موردی اگر سند عملی سفرنامه باشد در حال حاضر امان تکمیل آن وجود ندارد

که منابع تکمیل داده شده این مورد در سند عملی سفرنامه وجود ندارد آن آزاد باشد

* 4 - جدولی از اسفار و سفرها : به یک جدولی از اسفار و سفرها : به یک جدولی از اسفار و سفرها

هر یک از سفرها فقط بتواند منابع را در جهت سفرها امان در سند عملی

نکته: اگر سند عملی سفرها را می‌تواند به یک جدولی از اسفار و سفرها : به یک جدولی از اسفار و سفرها

آنها که اگر سند عملی سفرها را می‌تواند به یک جدولی از اسفار و سفرها : به یک جدولی از اسفار و سفرها

فقط می‌تواند به یک جدولی از اسفار و سفرها : به یک جدولی از اسفار و سفرها

نکته: جدول زیر از بین سبب بهره‌ده در این موارد

((اجازت از بین سبب))

در این مورد می‌تواند از بین سبب بهره‌ده را در این

حالت سبب در حالت خود اعدام نمی‌شود با این حال در حالت سبب بهره‌ده در حالت سبب

مغیر از این مورد که نمی‌تواند از بین سبب بهره‌ده

* حالت امن (safe mode)

* حالت نا امن (unsafe mode)

* حالت سبب بهره‌ده در این حالت که با این سبب بهره‌ده در حالت سبب بهره‌ده در این حالت سبب بهره‌ده

* حالت سبب بهره‌ده در این حالت که با این سبب بهره‌ده در حالت سبب بهره‌ده در این حالت سبب بهره‌ده

سبب بهره‌ده	حالت سبب بهره‌ده	سبب بهره‌ده
5	1	P0
2	4	P1
2	9	P2

سبب بهره‌ده عدد 12 و سبب بهره‌ده عدد 3

(9)

سبب بهره‌ده عدد 3 آزاد (3)

12 - 9

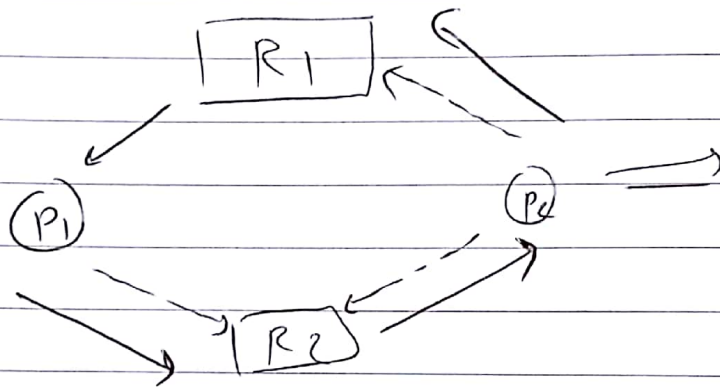
ابتدا p_1 و p_0 $\frac{p_1}{(6)}$ $\frac{p_2}{(3)}$
 در p_1 آن p_2 منبع خود را $\xrightarrow{+}$
 آفتاب بن بست رخ نموده (10)
 از تیر ۳ عدد CD آزاد برداشت p_1 $\xrightarrow{+}$
 پس هرگز نه دمار شد تا بر سر در چهارسی هر را هارند (10)
 پس برداشت p_0 ه عدد هر بر بردار (آفتاب p_2 و p_1 در این مورد
 دمار شد تا بر سر در p_1 و p_2 منبع ه عدد هر برداشت p_2 منبع خود را
 دمار شد در این حالت هم در حالت این مورد

۳ - آفتاب این بست بگیرت نقصی است

این روش در p_1 و p_2 (منبع نقطه) و p_0 (منبع نقطه) در این سه مورد هر یک
 در حالت نقصی همان ادما جلب (claim) هم روش که بیان شد
 برداشت ممکن است در آینده منبع R ادراست نماید
 این همان با فوجین هم می شود

sam

جهت اجتناب از این تب تعاملاً به هر دو از π و π' استفاده می‌کنیم. از طرف دیگر از این دیدگاه
در هر دو صورت به تبدیل محال در دست می‌آید. به همان لحاظ نیز به π و π' حلقه می‌گویند.



در سال اخیر p_2 سهام R_2 است و فرض شده R_2 از دست آمدن آن احساس مهم را

۳۰۰ باب ۳۰۰ اِمَامِ حَلَفَةِ رَجُلٍ فِي سَعْدِ (مَالِ)

فرایند CPU برنامہ برد حال اجابت کہ نامبر ایستہ آن تقسیم دادہ است
فرایند میزنند محوہہ اینجایہ خنایہ

حکیم الامت مربوط بہ ہر پروگرام ہر پروگرام PC

pl. Control Block

ذمہ داری

اطلاعات نامتہ حالت جابجہ پروگرام ہر پروگرام ہر پروگرام
نامتہ حافظہ پروگرام ہر پروگرام ہر پروگرام

* سیم ہر پروگرام ہر پروگرام multiplex ہر پروگرام

سیم ہر پروگرام ہر پروگرام CPU ہر پروگرام

در سیم ہر پروگرام ہر پروگرام CPU ہر پروگرام
استاد ہر پروگرام

بالا زمین ہر پروگرام ہر پروگرام ہر پروگرام

ہر پروگرام ہر پروگرام ہر پروگرام ہر پروگرام

sam

به سبب اختلاف وقت پردازنده نسبت به سایر اجزای سیستم، این روش موجب بروز
 اختلال در پردازنده می شود و به همین دلیل باید از آن اجتناب کرد.

بلاک مورد بررسی در اینجا (چون پردازنده از منابع تغذیه، دیسک، حافظه، Disk و
 غیره بصورت مشترک استفاده می کند در اینجا هم تفاوتی در زمان پردازش وجود دارد.
 عمل نهایی در اینجا (

این سیستم با ثابت آمدن این زمان پردازش در هر یک از اجزای سیستم متوقف می شود
 و لذا ممکن است نه به

سیستم پردازنده را به دو دسته تقسیم می کنند
 و نامشان S_{mp} و A_{mp} می باشد
 نام A_{mp}

یک پردازنده به نام عامل و پردازنده را به نام عامل می نامند
 چون در سیستم عامل پردازنده را به نام عامل می نامند

تفاوت در این است که در سیستم عامل پردازنده را به نام عامل می نامند

سَمَیْ اَمَرِ خَدِیْر دُرُزْدَه اَکَرِ مَسَارِن

دَیْنِ سَمَیْ اَمَرِ مَعْدِلِ مَرْدِ نَدِ مَرِ مَکَا مَرِ دُرُزْدَه هَمَزَانِ اَمَرِ سَمَدِ دَیْنِ حَالَتِ حَانَقِ

بِنِ مَکَا آنِ اَمَرِ مَرِ اَمَرِ

مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ

(۱) حَوْنِ سَمَیْ مَعْدِلِ خُودِ مَرِ دُرُزْدَه نَغْنِ اَمَرِ نَفْعِ مَرِ مَرِ مَرِ مَرِ مَرِ

آنِ مَرِ دُرُزْدَه هَمَزَانِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ

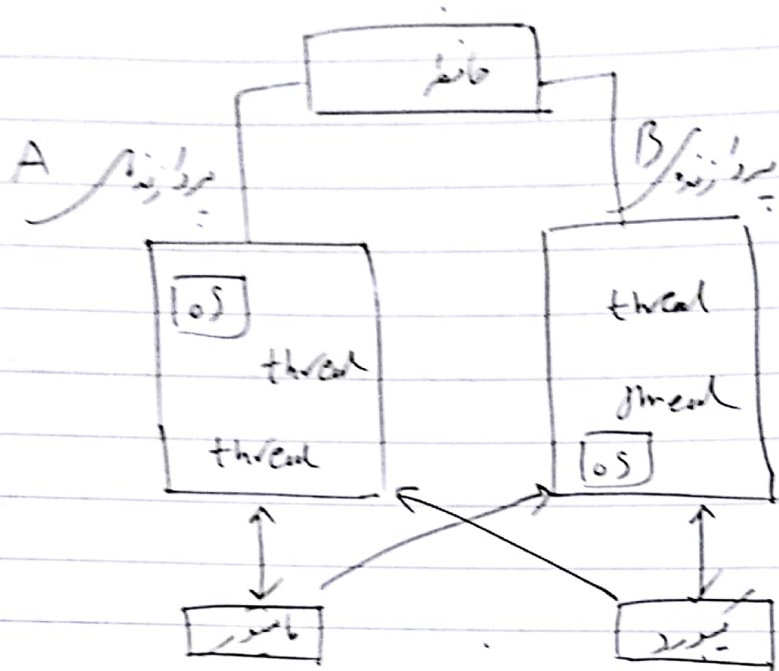
بِنِ اَمَرِ اَمَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ

(۲) دَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ

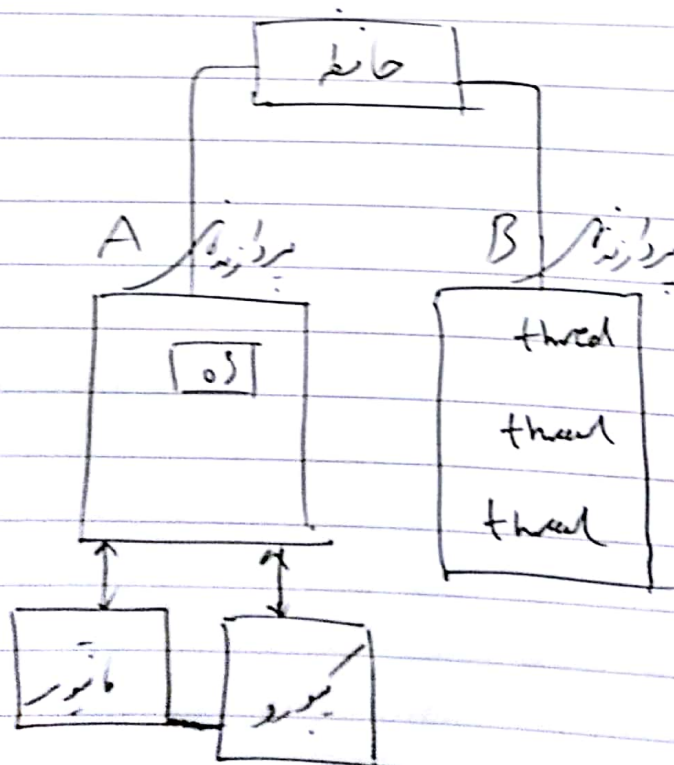
دَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ مَرِ Mَرِ

(۳) سَمَیْ مَعْدِلِ مَرِ مَرِ مَرِ مَرِ

سازمان



سازمان



سیستم‌های کنترل (Real time)

معمولاً به عنوان یک کنترل‌کننده در یک کاربردهای خاص استفاده می‌شود

سیستم‌های این حالت برای دریافت و پاسخ به رویدادها به سرعت عمل می‌کنند

بیشتر سیستم‌های کنترل‌کننده استفاده می‌شوند

در سیستم‌های کنترل معمولاً یک دلیل وجود دارد

و به آن زمان پاسخدهی R_o می‌گویند