سیستم عامل را به دو صورت میتوان تعریف نمود

سیستم عامل استفاده از کامپیوتر را ساده میسازد ،بدین معنی که کاربر یا برنامه نویس براحتی میتواند با کامپیوتر کار کند،در صورت عدم و جود سیستم عامل کاربر یا برنامه نویس میبایست اطلاعات کاملی از سخت افزار کامپیوتر (cpu,ram,hard,IO,...) داشته باشد.

وظیفه دوم سیستم عامل (Resource Management)مدیریت منابع می باشد،بدین معنی که سیستم عامل باعث استفاده بهینه از منابع فیزیکی میگردد.منابع فیزیکی سخت افزار کامپیوتر می باشد و منظور از منابع منطقی فایلها و داده های ما هستند.

از دید (بالا به پایین) سیستم عامل واسطی راحت را برای کاربران فراهم می سازد و از دید(پایین به بالا) سیستم عامل مدیر کلیه منابع سیستم می باشد.

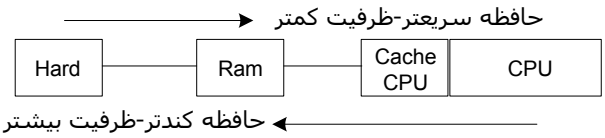
Resident

اولین برنامه ای که بعد از روشن شدن کامپیوتر بالا می آید سیستم عامل نام دارد و بعد از باز شدن سیستم عامل قسمتی از آن در حافظه بصورت دائم باقی می ماند که به آن Resident گویند.

Transient

قسمتی دیگر از سیستم عامل بصورت دائم در حافظه وجود ندارد و بین حافظه و دیسک در حرکت است که به آن برنامه ها Transient گفته می شود.

سلسله مراتب حافظه



چرا در کامپیوتر ازسلسله مراتب حافظه استفاده میشود؟

ایده آل ما در کامپیوتر حافظه ای است که سرعت بالا(در حد cpu) ،ظرفیت ذخیره سازی بالا(مانند hard) ، و با قطع جریان برق اطلاعات آن از بین نرود ولی چنین حافظه ای در حال حاضر موجود نمیباشد ، بنابراین از مجموعه ای از حافظه ها استفاده میکنیم که هر کدام مزایا و معایبی دارد.

Hard disk

مزایا:

با قطع جریان برق اطلاعات از بین نمیرود ،ظرفیت بالا

معایب:

سرعت پایین در انتقال اطلاعات

Ram,Cache

مزایا:

سرعت از هارد خیلی بالاتر است

معایب:

با قطع جریان برق اطلاعات از بین میرود ،ظرفیت پایین نسبت به هارد

سرعت cache از ram بیشتر است و ظرفیت ان از ram پایینتر

- اگر کامپیوتری Ram نداشته باشد اصلا کار نمیکند.
- اگر کامپیوتری هارد نداشته باشد میتوان سیستم عامل را از رسانه های ذخیره سازی دیگر به درون Ram انتقال داد و سیستم در اینصورت کار میکند.

معماری سخت افزار کامپیوتر همزمان با رشدش تحولی را در سیستم عاملها بوجود آورده است ،از اینرو نسلهای مختلف سیستم عامل را با یکدیگر بررسی میکنیم.

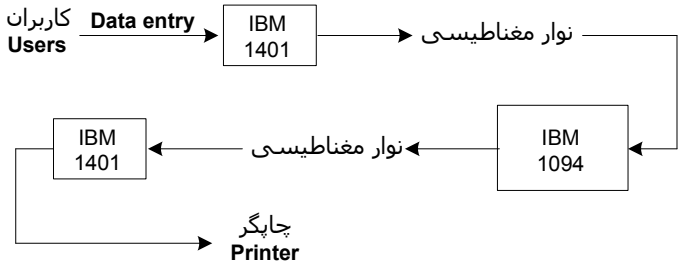
1945-1955 نسل اول (1)
Punch card

در این نسل از لامپ خلاء برای ذخیره سازی اطلاعات استفاده میشد و زبان برنامه نویسی اسمبلی ابداع نشده بود و سیستم عاملی نیز وجود نداشت و برنامه نویسان تنها در یک فاصله زمانی کم حق استفاده از کامپیوتر ها را داشتند و برنامه های خود را بر روی کارت پانچ ها وارد کرده و به کامپیوتر می دادند.

1955-1965 نسل دوم (2)
Batch (دسته ای)

در این نسل ترانزیستورها مطرح شدند و برنامه ها بصورت دسته ای اجراء میشدند، بدین صورت که چندین برنامه را باید بصورت مشترک به کامپیوتر می دادیم و کامپیوتر بعد از اجرای تمامی آن برنامه ها خروجی را به ما تحویل میداد و زبانهای برنامه نویسی اسمبلی و فرترن بودند.

Offline Spooling



در سیستمهای دسته ای مشکل بدین صورت بود که کامپیوتر قدرتمندی مانند IBM 1094 می بایست زمان زیادی را صرف عملیات ورودی و خروجی برنامه کند(وارد کردن برنامه ها)

در تکنولوژی offline spooling روش کار بدین صورت است که کامپیوتر ضعیفی مانند IBM1401 را مسئول دریافت داده ها و برنامه های کاربری میکردند که ورودی با کیبورد صورت میگرفت و خروجی آن در نوار مغناطیسی به کامپیوتر قدرتمندی مانند IBM1094 داده میشد، کامپیوتر قدرتمندبرنامه های روی نوار را با سرعت زیادی اجراء کرده و خروجی را بر روی نوار به کامپیوتر ی با قدرت کمتر IBM1401 می داد وآن کامپیوتر خروجی را بر روی کاغذ پرینت میکرد.

معایب:

1- زمان گردش کار طولانی

2- عدم وجود اولویت در اجرای برنامه

3- نیاز به سخت افزار اضافی

مزایا:

1- افزایش راندمان سیستم(کارائی)

2- امکان استفاده از راه دور

در محلی خارج از کامپیوتر قوی با یک کامپیوتر ضعیف نوارها آماده میشد.

نسل سوم (3)
سیستمهای چند برنامه ای

در این نسل از مدارات مجتمع استفاده میشود.و برنامه ها به دو دسته کلی تقسیم میگردد.

I/O Limited :1

با محدودیت ورودی و خروجی (برنامه نیاز به ورودی و خروجی زیاد دارد)

برخی از برنامه ها در روند اجراء شدن زمانهای زیادی را صرف گرفتن داده از ورودی و یا نوشتن داده در خروجی میکردند،بدیهیست که اجرای چنین برنامه ای توسط کامپیوتر قدرتمندی مانند IBM1094 باعث میشد که آن کامپیوتر قدرتمند زمانهای زیادی بیکار بماند و پردازش زیادی انجام ندهد.

CPU Limited :2

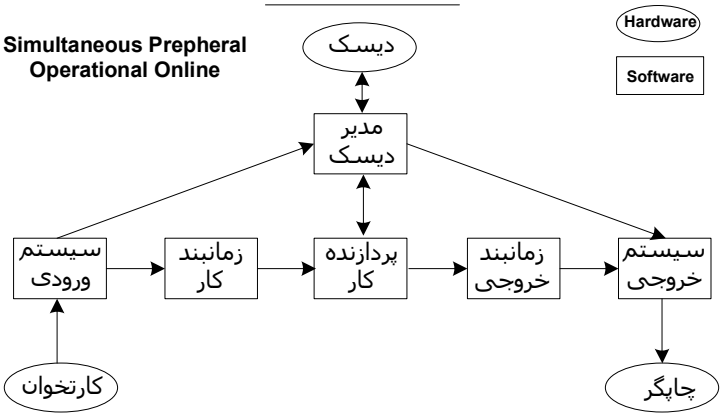
با محدودیت پردازش (برنامه نیاز به کار پردازشی زیاد دارد)

برخی از برنامه ها ماهیتا بیشتر کار پردازشی میکنند تا درخواست ورودی و خروجی اجرای چنین برنامه ای توسط کامپیوتر باعث این میشود که پردازنده کارائی بالایی داشته باشد.

یکی از ایرادهای سیستمهای دسته ای این بود که اگر کار I/O limited را به کامپیوتر می دادیم cpu زمان زیادی رابیکار می ماند ،در نتیجه برای بهبود چنین مشکلی سیستمهای چند برنامه ای مطرح شدند که هدف آنها اجرای همزمان چندین برنامه بود که این برنامه ها میتوانستند از نوع i/o limited ویا cpu limitedباشند.

Bay
www.freebay.ir

Online Spooling



در چنین سیستمی که به online spooling مطرح است عمل i/o یک کار را با عمل محاسباتی کار دیگر ،روی هم می اندازد(overlap) روند چنین فعالیتی در سیستم spooling به این صورت است که کاراکترهایی که توسط کارتخوان ،خوانده میشوند در بلوکهایی جمع بندی شده و توسط مدیر دیسک بر روی دیسک ذخیره میشود و در انتهای این عملیات سیستم ورودی اطلاعاتی راجع به آن کار را به زمانبند کار ارسال میکند،که این اطلاعات شامل نام کار،محل قرارگیری کار در دیسک و اولویت کار می باشد.

زمانبند کار ،یک لیست از کارهای موجود در ماشین و اطلاعات لازم برای هر کار را در صفی نگهداری میکند و به پردازنده کار میگوید که کار بعدی را اجراء کند و اطلاعات آن کار را که شامل محل فیزیکی آن کار ،نام آن و اولویت آن است را به پردازنده کار تحویل میدهد.

پردازنده کار در واقع همان بخشی است که برنامه را اجراء میکند ،این پردازنده محل اجرای آن کار را بر روی دیسک میداند ،بعد از اتمام عمل پردازش پردازنده کار خروجی خود را بصورت بلوکهایی بر روی دیسک مینویسد و مدرکی را به زمانبند خروجی تحویل میدهد.

زمانبند خروجی لیستی از برنامه های منتظر چاپ را در اختیار دارد که قرار است بلوکهای خروجی را از مدیر دیسک به سیستم خروجی تحویل دهد که ترتیب انتقال خروجی کارها از مدیر دیسک به سیستم خروجی توسط زمانبند خروجی صورت میگیرد.

سیستم خروجی بلاکها را کاراکتر به کاراکتر به چاپگر ارسال میکند .

مزایا:

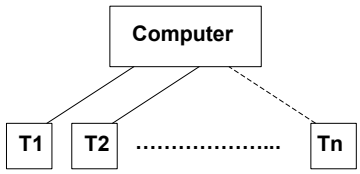
1- اختصاص اولویت به کارها
2- عدم نیاز به سخت افزار اضافی
3- کاهش زمان گردش کار(چرخه اطلاعات زمانبری کمتر دارد)

4 نسل چهارم — سیستمهای اشتراک زمانی
Time sharing

این سیستمها در واقع توسعه یافته سیستمهای چند برنامه ای بودند سیستمهای چند برنامه ای کاربر ارتباطی با کامپیوتر نداشت و اشکال زدایی برنامه ها کار دشواری بود و زمان برگشت خروجی طولانی بود

هدف از ایجاد سیستمهای اشتراک زمانی ، سیستمهایی با محاوره (intractive) بالا بود.

در این سیستم کاربرها به کمک ترمینالهایی به کامپیوتر متصل میشوند ،این ترمینالها دارای یک مانیتور و یک کیبورد بود،کاربر مستقیما دستوراتی را با کیبورد وارد کرده و خروجی را بر روی مانیتور مشاهده میکرد، در سیستم اشتراک زمانی تنها یک پردازنده وجود دارد که بین ترمینالها به اشتراک گذاشته شده است که اشتراک گذاری توسط مکانیزمهایی صورت میگیرد که با سرعت زیاد cpu بین برنامه ها سوئیچ میکند.
از این نسل ،سیاست در بکارگیری نرم افزارها تغییر کرده است ،یعنی سیاست افزایش کارایی سیستم جای خود را به راحتی کاربر داده است،دراین سیستمها این امکان فراهم است که بتوان قدم به قدم برنامه ها را اشکال زدایی کرد(debug)



5 سیستم عامل کامپیوترهای شخصی

این کامپیوتر ها جزء نسل چهارم میباشند ،با ایجاد مدارهای مجتمع و توسعه آنها استفاده وسیعتر از کامپیوتر امکان پذیر شد.رفته رفته کاربرد آنها بصورت شخصی مطرح گردید، در ابتدا سیستم عاملها بصورت تک کاربره و تک برنامه ای بودند مانند سیستم عامل DOS و رفته رفته قابلیت چند برنامه گی و چند کاربره بودن در چنین سیستمهایی مطرح گردید.

بعلت کاهش هزینه سخت افزار ،سیاست در چنین سیستمهایی راحتی کاربر و نه ماکزیمم کارایی سیستم بود.
هنگامی که چند کامپیوتر شخصی به هم متصل میشوند به آنها worksataion گویند.

6 سیستمهای توزیع شده
Distributed system

این سیستمها در یک محیط شبکه ای قابل اجراء هستند،در این سیستم قسمتهای مختلف یک برنامه بدون اینکه کاربر متوجه شود بین چندین سیستم تقسیم میشود و برنامه کاربر بر روی چندین سیستم بصورت همزمان اجراء میشود که باعث افزایش کارایی اجرای برنامه میگردد.

سیستم عاملهای توزیع شده بسیار پیچیده تر از سیسم عاملهای قبلی هستند

مزایا:

1-سرعت بالای اجرای برنامه ها
2-حالت Transparent :کاربر متوجه حالت distribute نمیشود.

7 سیستمهای چند پردازنده
Multi processor

در سیستم چند پردازنده ای کامپیوترها بجای یک cpu از چندین cpu بهره میبرند.در این سیستمها چند پردازنده از یک حافظه اصلی ،ورودی خروجی یکسان و رسانه ذخیره سازی واحد بهره میبرند .

مزایا:

1- افزایش توان عملیاتی سیستم(کارایی)

2- تحمل پذیری خطا
در صورت خراب شدن یکی از پردازنده ها امکان اجرای برنامه ها توسط برنامه دیگر امکان پذیر است بحالتی که سیستم با کارائی کمتر به کارخود ادامه میدهد(تنزل مطلوب)

3- کاهش هزینه

سیستمهای چند پردازنده ای به دو دسته تقسیم میشوند.
Symmetric-Asymetric

Symetric
متقارن
در این سیستم پردازنده ها بصورت یکسان پیاده سازی میشوند و برنامه ها از هر گروهی که باشند میتوانند در هر پردازنده ای اجرا شوند.
Solaris-win nt

Asymetric
نامتقارن
یک پردازنده وظیفه اجرای دستورات سیستم عامل را بعهده میگیرد و پردازنده های دیگر برنامه های کاربر را اجراء میکنند.

معایب:

1- فعالیت یک پردازنده(اجراء کننده دستورات سیستم عامل)خیلی بیشتر از پردازنده های دیگر میباشد.

Bay
www.freebay.ir

2- اگر پردازنده اجرای دستورات سیستم عامل خراب شود کل سیستم از کار می افتد.
3- سیستمهای نا متقارن غیر قابل حمل هستند ،بدین مفهوم که باید روی هر سخت افزار نامتقارن سیستم عامل منحصر به فردی را پیاده سازی نمود مانند سیستم عامل Sun os V4

(8) سیستمهای بلا درنگ
Realtime system

سیستمهای بلادرنگ به دو دسته تقسیم میشوند.
Hard realtime-Soft realtime

Hard realtime

بلادرنگ سخت

در این سیستم باید عملیات مورد نظر در یک زمان محدود انجام گردد.وعدم اجرای برنامه در آن زمان غیر قابل قبول است

در چنین سیستمهایی نباید واسط نرم افزاری(مانند سیستم عامل)بین برنامه کاربردی و سخت افزار وجود داشته باشد.مانند پرنامه کنترل کننده یک موشک غیر بالستیک(غیر پرتابی) یا برنامه کنترل کننده روبات

در این نوع سیستم رسانه ذخیره سازی وجود ندارد و همچنین برنامه در حافظه ای مانند Rom نوشته شده و مورد استفاده قرار میگیرد.

Soft realtime

بلادرنگ نرم

در سیستم بلادرنگ نرم کار باید در یک محدوده زمانی اجراء شود ولی اگر برنامه در آن محدوده زمانی اجراء نشد اتفاق غیر قابل جبرانی نمی افتد مانند رزرواسیون هواپیما –مالتی مدیا

ساختار(مولفه های) سیستم عامل

بطور کلی خدمات و مولفه هایی که سیستم عامل ارائه میدهد به بخشهای زیر تقسیم میشود.

1- مدیریت پردازنده
2-مدیریت حافظه اصلی
3- مدیریت حافظه مجازی

حافظه اصلی * 1.5 =حافظه مجازی

4-مدیریت فایل
5- مدیریت وسایل I/O
6-مدیریت منابع نرم افزاری

(1) مدیریت پردازنده
Process managment

این برنامه شامل مجموعه ای از دستورالعملها و داده ها میباشد که در هنگام اجرای دستورالعملها به آن نیاز است.

برنامه یک ماهیت غیر فعال دارد(passive) و هنگامی که پردازنده ای به برنامه اختصاص میابد برنامه که بر روی هارد قرار دارد و مانند یک فایل نگهداری میشود به حافظه اصلی انتقال میابد و به این حالت که به برنامه حافظه اصلی و cpu اختصاص داده شد پردازش (process) گویند.

پردازش (process) یک ماهیت فعال (active) است .منابعی که یک process برای اجراء شدن نیاز دارد شامل cpu ،حافظه اصلی،وسایل I/O و فایلها میباشد.

وظائف سیستم عامل در رابطه با مدیریت پردازنده

1- ایجاد و حذف پردازشهای کاربر و سیستم
2- زمانبندی پردازشها و اینکه در هر زمان پردازنده به کدام فرآیند(process) داده شود.
3-مدیریت همزمانی پردازشها و یا مدیریت بن بستها(dead lock)

(2) مدیریت حافظه اصلی
Ram managment

Cpu بطور مستقیم با حافظه اصلی در ارتباط است ،برنامه ها نیز برای اینکه به فرایند تبدیل شوند باید در حافظه اصلی قرار گیرند.

در سیستمهای چند برنامه ای نیاز است که چندین فرایند بصورت همزمان در حافظه قرار گیرند،مدیریت قرار گیری برنامه ها در حافظه بعهده سیستم عامل است،تخصیص و بازپذیری فضای حافظه به فرایندها (process) نیز بر عهده سیستم عامل است.

(3) مدیریت حافظه مجازی
Virtual memory managment

از آنجائیکه Ram برای جا دادن تمامی برنامه ها فضای کافی ندارد از اینرو سیستم عامل وظیفه دارد قسمتی از حافظه ثانویه(Hard) را تحت عنوان حافظه Ram معرفی کرده که به آن حافظه،حافظه مجازی گویند.

(4) مدیریت فایل
File system managment

برای ایجاد یک دید منطقی واحد از اطلاعات ذخیره شده بر روی انواع رسانه ها از قسمتی از سیستم عامل بنام مدیریت فایل(file system) استفاده میشود.

وظائف سیستم عامل در رابطه با مدیریت فایل

1- ایجاد و حذف فایلها
2- ایجاد و حذف دایرکتوریها (Folders)
3- عملیات copy وcut کردن فایلها و دایرکتوریها
4-مدیریت دسترسی های مختلف به فایلها بصورت مشترک

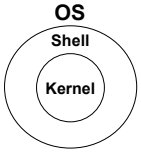
(5) مدیریت وسایل ورودی و خروجی
I/O Managment

هر وسیله I/O حافظه ای بنام buffer دارد که اطلاعات خوانده شده و نوشته شده در آن قرار میگیرداز وظائف سیستم عامل جلوگیری از تداخل وسایل مختلف با یکدیگر است و جلوگیری از بن بست ها نیز از دیگر وظائف مدیر I/O می باشد.

(6) مدیریت دیگر منابع

مدیریت حسابهای کاربری،تشخیص خطا،مدیریت نرم افزار،....وظائف دیگر سیستم عامل می باشد.

مفسر فرمان



قسمتی از سیستم عامل که واسط بین کاربر و سیستم عامل میباشد را مفسر فرمان گویند.مفسر دستورات را از کاربر گرفته و به قسمتهای مختلف سیستم عامل ارجاع میدهد .

در بعضی سیستم عاملها قسمت مفسر فرمان در هسته(kernel) سیستم عامل پیاده سازی شده است و در برخی دیگر در پوسته(shell) قرار گرفته است.قسمت مفسر فرمان در سیستم عامل میتواند یک واسط متنی یا یک واسط گرافیکی باشد.

Dos وUnix واسط مفسر فرمان متنی دارند و Windows وLinuxمفسر فرمان گرافیکی دارند.

تشخیص اتمام انتقال داده

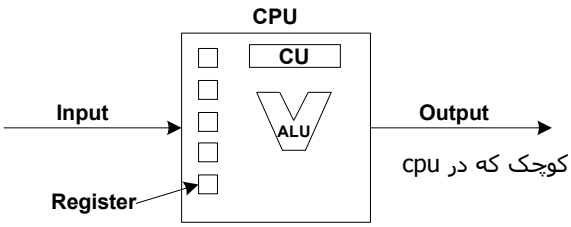
در سیستمها به دو صورت میتوان تشخیص داد که انتقال داده از قسمتی از ابزارهای I/O به حافظه Ram صورت گرفته است.

1- روش سرکشی Polling
2- روش وقفه Intrrupt

(1) روش سرکشی
Polling

در این روش هر وسیله I/O ثباتی (Register) درون cpu دارد و cpu وظیفه دارد در هر پریود زمانی تمامی ثباتهای کنترلی I/O های مختلف را بررسی کند(سرکشی کند). و در صورت اتمام انتقال داده به عملیات بعدی بپردازد ،این روش مانند کلاس درسی میماند که استاد هر چند دقیقه یکبار از تمامی دانشجویان بپرسد که آیا سئوالی دارند یا نه؟

Bay
www.freebay.ir



ثبات:حافظه ای کوچک که در cpu قرار دارد

روش وقفه Intrrupt (2)

در این تکنیک هر وسیله I/O دارای سیگنال کنترل مخصوص به خود میباشد که به نحوی با cpu در ارتباط است هر گاه انتقال داده توسط آن دستگاه تمام شد سیگنالی را به نام وقفه به cpu ارسال میکند در این حالت پردازنده فعالیتش را متوقف کرده و به سرویس دهی این وقفه می پردازد،این روش مانند کلاس درسی است که هر گاه دانشجویی سئوالی داشت با بلند کردن دست خود به استاد اطلاع دهد.

روش وقفه روش کارامدتری است و سیستم عاملها از آن استفاده میکنند.

انواع وقفه ها

1- وقفه برنامه: **Program check**

این وقفه هنگامی اتفاق می افتد که برخی از دستورات رخ دهد،در برخی از مواقع به این وقفه تله(Trap)می گویند.بعنوان مثال وقتیکه در برنامه عملیات تقسیم بر صفر Divide by zero صورت میگیرد و برنامه وقفه از نوع فوق صادر میکند.

2- وقفه زمانسنج: **Timer**

برخی از فعالیتها در سیستم عامل بصورت پریودیک انجام میشود،مانند تنظیم ساعت،چک کردن سخت افزارها،.....اینها از وقفه های تایمر استفاده میکنند.

3- وقفه I/O:

این وقفه توسط دستگاه ورودی و خروجی تولید میشود تا کامل شدن عمل انتقال را اطلاع دهد.

4- وقفه نقص سخت افزار:

در این وقفه اگر اتفاقی از لحاظ سخت افزاری رخ دهد تولید میشود بعنوان مثال نقص برق

5- وقفه فراخوان سوپروایزر: **(SVC) Supervisor call**

اگر درخواست خارج از سرویس از طرف کاربر صادر شود وقفه ای از این نوع تولید میشود.

6- وقفه **Restart** :

وقفه ای که با فشار دادن کلید Restart تولید میشود.

انواع وقفه ها از جنبه دیگر

1- وقفه داخلی یا Trap :

وقفه هایی هستند که از دستورات خود برنامه تولید میشوند،مانند گروه اول وقفه های دسته بندی قبلی

2- وقفه های خارجی:

این وقفه ها از دستگاههای خارجی تولید میشوند مانند وسایل I/O ،تایمرها ،خطاهای سخت افزاری و.....

3- خطاهای نرم افزاری یا SVC :

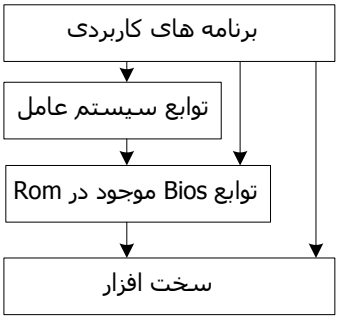
این وقفه همانند وقفه SVC در دسته بندی قبل میباشد.

انواع سیستمهای عامل از نظر سخت افزاری

سیستمهای یکپارچه (1)

در این نوع سیستم عاملها ،سیستم عامل بصورت مجموعه ای از رویه ها procedure نوشته شده است که هر یک از رویه ها در صورت نیاز میتوانند دیگری را فراخوانی کنند،در این معماری هیچ مکانیزمی برای مخفی کردن رویه ها وجود ندارد و تمامی رویه ها قابل مشاهده هستند .مثالی که از این سیستم عامل میتوان زد سیستم عامل DOS میباشد.

نمودار زیر شمای کلی این نوع سیستم عامل را نشان میدهد.



اکثر cpuها دارای دو مد کاری هستند.
۱ - مد هسته
۲ - مد کاربر

مد هسته:
مخصوص سیستم عامل است و در این مد تمامی دستورالعملها مجاز است.

مد کاربر:
مخصوص برنامه های کاربردی است و در این مد برخی از دستورات،دستورالعملهای غیر مجاز میباشد .مانند دستورات I/O

سیستم عامل DOS توسط پردازنده زمان خود (8086-8088) محدود بوده است و این پردازنده تنها در یک مد کار میکرد

پردازنده های 386 پردازنده هایی بودند که در 2 مد کار میکردند.

سیستمهای لایه ای (2)

در سیستمهای لایه ای سیستم عامل را به چند سطح یا لایه تقسیم میکنیم،که هر کدام از لایه ها در بالای لایه پایینتر قرار میگیرد ،مزیت مهم این سیستمها پیمانه ای بودن آنهاست(modularity) ،بدین معنی که لایه ها به گونه ای تقسیم بندی میشود که تنها بتواند رویه های لایه های پایینتر را سرویس دهی کند ،از اینرو میتوان هر لایه را بصورت جداگانه طراحی نمود.

اولین سیستم لایه ی سیستم عامل THE با 6 لایه بود بشرح زیر:

5	System operator
4	User program
3	I/O Managment
2	Comunicator
1	Memory managment
0	CPU Timing

لایه شماره 0:
بدین معنی که CPU هر لحظه چه برنامه ای را اجرا کند.

لایه شماره 1:
وظیفه مدیریت حافظه

لایه شماره 2:
برقراری ارتباط بین هر process و کنسول اپراتور

لایه شماره 3:
مدیریت وسایل ورودی و خروجی

لایه شماره 4:
برنامه های کاربر

لایه شماره 5:
اپراتور سیستم

در مدل لایه ای مشکل اصلی تعریف مناسب لایه های مختلف است.

معایب:

1- از آنجائیکه هر لایه میتواند فقط با لایه های پایینتر از خود کار کند ،برای انتخاب لایه ها نیاز به دقت فراوانی است.

Bay
www.freebay.ir

2- بازدهی کم آن نسبت به روشهای دیگر

برای اینکه دستورات از یک لایه به لایه ای دیگر انتقال یابد نیاز است دستورات مورد بررسی قرار گیرد که این عمل به سرباری (overhead) سیستم می افزاید

نسخه های اول WIN NT با لایه های زیاد ایجاد شد،در WIN 95 تعداد لایه ها کمتر شد و بهمین صورت در WIN NT4 سعی گردید تعداد لایه ها کمتر و بصورت مجتمع تری پیاده سازی شود تا کارائی سیستم افزایش یابد.

③ ماشین مجازی VM(virtual machine)

سیستم عامل VM برروی IBM میتواند بهترین مثال از نمونه های ماشین مجازی باشد.

نمودار زیر ساختار کلی یک ماشین مجازی را نمایش میدهد.

برنامه کاربر 1	برنامه کاربر 2	برنامه کاربر 3
CMS	**CMS**	**CMS**
مانیتورماشین مجازی		
سخت افزار		

CMS=Conversation Monitor System

قلب سیستم که به مانیتور ماشین مجازی معروف است برروی سخت افزار عریانی اجراء میشود و قابلیت چند برنامه گی را فراهم می آورد.

این مانیتور چندین ماشین مجازی را در لایه های بالاتر فراهم می سازد که هر ماشین مجازی مشابه یک نسخه از سخت افزار عریان است که شامل (حافظه،پردازنده،I/O .....) میباشد.CMS برای هر کاربر یک سیستم عامل تک کاربره و محاوره ای میباشد.

مزایا:

1- در این سیستم دو وظیفه اصلی چند برنامه گی و ایجاد واسط کاربر مناسب فراهم است.

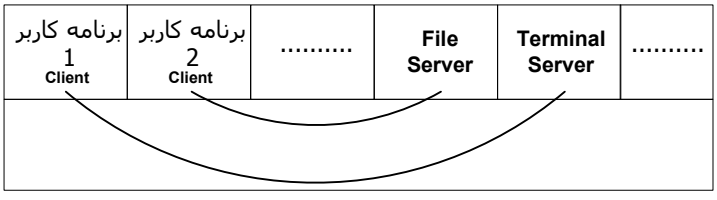
2- هر ماشین مجازی از سایر ماشینها مجزا است،بنابراین هیچ مشکلی از لحاظ امنیتی وجود نخواهد داشت و سیستم عاملها تداخلی با یکدیگر ندارند.

④ سیستمهای Client/Server

سیستمهای ماشین مجازی بخش زیادی از کدشان در لایه بالاتر (CMS)اجرا میکردند،با اینکه هسته سیستم عامل (مانیتور ماشین مجازی) هسته نسبتا ساده ای بود ولی دارای پیچیدگیهای زیادی بود.

این ایده که سیستم عاملها جوری طراحی شوند که قسمت هسته آنها سبکتر و ساده تر شود و فعالیتها را به لایه های بالاتر انتقال دهیم باعث افزایش کارائی سیستم عامل میگردد.از اینرو بهتر است

اگر وظائف سیستم عامل در لایه کاربران یا برنامه های کاربردی پیاده سازی شود معماری به شکل زیر طراحی میگردد.



بعنوان مثال برای خواندن یک بلاک، فایل process کاربر یک درخواست به سرور ارسال میکند و از آن میخواهد که کارش را انجام دهد.در چنین معماری تنها کاری که هسته انجام میدهد ارتباط بین پردازش client و پردازش server میباشد.

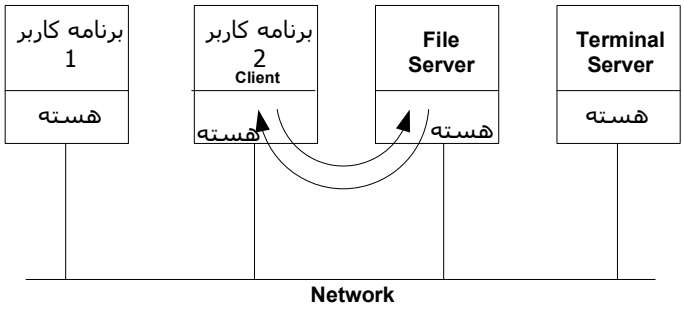
مزایا:

1- از آنجا که سیستم عامل به چندین بخش تقسیم میشود طراحی آن ساده تر است.

2- بعلت اینکه فعالیتها در برنامه های process های بین clientوserverاجرا میشود،یا به عبارت دیگر این process ها در مد کاربر اجراء میشوند و نه در مد هسته هیچکام از آنها نمیتوانند بصورت مستقیم با سخت افزار ارتباط برقرار کنند.

3- چنین سیستمی قابلیت پیاده سازی برای سیستمهای توزیع شده را داراست.

معماری توزیع شده **Client/Server**



به حالت رفت وبرگشت در شکل Message Passing گویند.

مکانیزم و سیاست Mechanism & Policy

یک اصل مهم در پیاده سازی سیستم عاملها جداسازی سیاست از مکانیزم میباشد.

مکانیزم:چگونگی انجام کاری را نشان میدهد.

سیاست:آنچه را که باید انجام شود مشخص میکند.

بعنوان مثال اینکه چگونه cpu به پردازشی داده شود مربوط به مکانیزم است ولی اینکه چه مدت آن پردازش cpu را در اختیار بگیرد سیاست میباشد.

سیاستها احتمالا از ماشینی به ماشین دیگر متفاوت است ،در بدترین حالت با تغییر یک سیاست تغییری در مکانیزم مربوطه اش رخ میدهد.ولی بهترین حالت یک تغییر در سیاست باعث تغییر تنها چند پارامتر در مکانیزم میگردد.

زبانهای پیاده سازی سیستم عاملها

سیستم عاملهای اولیه با زبان اسمبلی نوشته میشد.ولی اکثر سیستم عاملهای امروزی با زبانهای سطح بالایی چون C++ نوشته میشود مانند Unix و یا Visual C++ برای Linux

مهمترین مزیت استفاده از زبانهای سطح بالا قابلیت حمل برنامه میباشد و همچنین سادگی در پیاده سازی و تغییرات در سیستم عامل از دیگر مزایای پیاده سازی آن با زبانهای سطح بالا میباشد.

اگرچه سیستم عاملها برنامه های بزرگی هستند ولی تنها قسمتی از کارهای سیستم عامل نسبت به کارائی، بحرانی است مانند مدیریت پردازنده،مدیریت حافظه اصلی،و.....

لذا بعد از پیاده سازی کامل سیستم عامل با زبان سطح بالا، قسمتهای بحرانی را با روالهای زبان اسمبلی بهینه ،جایگزین میکنیم.

پردازش و زمانبندی

پردازش یک ماهیت activeمیباشد،بعبارت دیگر مهمترین مفهوم در هر سیستم عامل فرایند یا پردازش میباشد.یک پردازش برنامه در حال اجراست به بیان بهتر هر پردازش process برنامه اجرائیست که علاوه بر کدهای برنامه (code segment)شامل مقدار شمارنده برنامه،رجیستر های cpu،پشته (stack) و بخشهای داده ای (data segment) می باشد.

در بیانی دیگر هر پردازشcpu مجازی خود را دارد،در حالت فنی بدین صورت است کهcpu در بخشهای زمانی کوتاهی بین برنامه ها به اشتراک گذاشته میشود.در طرف مقابل یک برنامه یک ماهیت passive است که شامل الگوریتمهایی است که درون یک فایل برروی هارددیسک ذخیره شده است.

پردازشها در دو دسته مختلف قابل اجراست.

1-پردازشهای مربوط به کاربر که دستورات معمولی را اجراء میکندو همچنین سطح دسترسی آن به منابع کم است.

2-مد سوپروایزر:که دستورات در این مد قابل اجراء هستند.

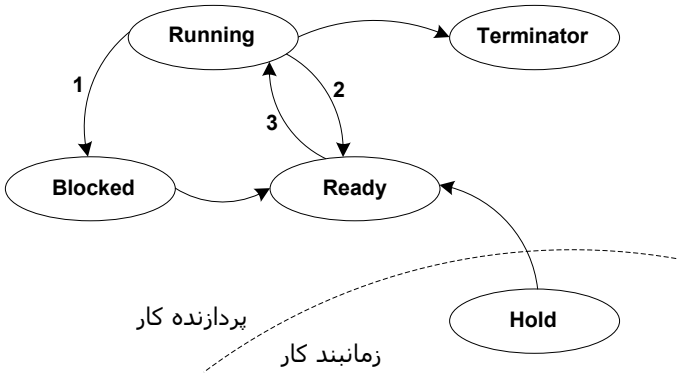
Bay
www.freebay.ir

زمانبند کار: **Job Schaduler**

روتینی است که بر اساس الگوریتمهای خاص برنامه های موجود در ISPT Input SPooling Table را انتخاب کرده و جهت اجرا شدن به حافظه منتقل میکند .در اینحالت آن برنامه باید تمامی منابع لازم برای اجراء شدن را در اختیار داشته باشد.

هنگامیکه کارها در جدول ISPT منتظر اجراشدن هستند (در واقع ISPT در دیسک قرار داردو هنوز به پردازش تبدیل نشده است) اصطلاحا در وضعیت Hold قرار دارند.

وقتی که توسط زمانبند کار جهت اجرا شدن انتخاب میشوند اصطلاحا به حالت Ready منتقل میشود.



الگوریتمهای زمانبند کار برای انتقال برنامه ها از حالتHold به حالت Ready عبارتند از(الگوریتمهایی که برنامه های منتظراجرا موجود درISPT به پردازنده کار تحویل میشود)

1. **FIFO: First in First out**
2. **SJF: Shortest job First**

برنامه هایی که برای اجرا شدن به زمان کوتاهتری نیاز دارند اول اجرا میشوند

3. **MIXED:**

در این روش ترکیبی از برنامه هایi/o limited وcpu limitedبرای اجرا شدن تحویل پردازنده کار میگردد.

حالات مختلف که یک پردازش میتواند در آن قرار بگیرد

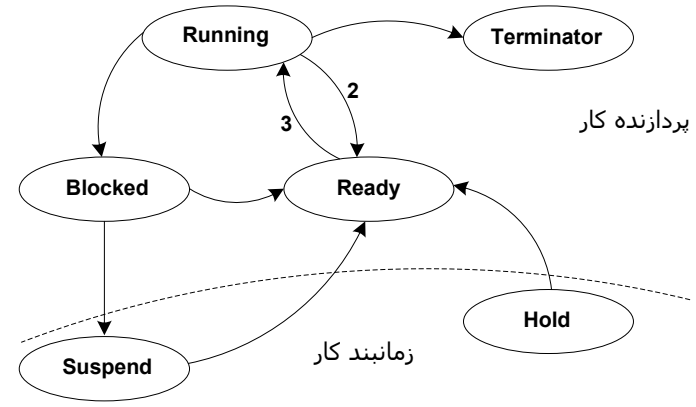
این حالات برای تمامی سیستمها مشترک است ولی برخی از حالات هستند که در سیستمهای دیگر مورد استفاده قرار میگیرند ،در ابتدا این حالات را برای یک سیستم عامل عمومی در نظر میگیریم

یک پردازش میتواند در سه حالت اجرا Running،آمادهReady،بستهBlock قرار گیرد.

هنگامیکه دستور اجرا برای پردازشی صادر میشود و یا زمانبند کار برنامه ای را از ISPT برای اجرا انتخاب میکند ابتدا درون صفReady وارد شده و منتظر گرفتن cpu میماند،قسمتهای 2و3 زمانبند پردازش process scheduler انجام میشود بدون آنکه خودprocess ازآن اطلاعی داشته باشد.

در انتقال3 cpu به پراسسی برای اجراء شدن تحویل داده میشود.در زمانها ی معینی که به آن برش زمانی گفته میشود cpu (Time Splice) از پردازشی گرفته شده و به پردازش دیگری داده میشود.
هنگامیکهcpu از پردازشی گرفته میشود آن پردازش به حالتReady میرود.

در حالت Ready پردازش،تمامی منابع مورد نیاز برای اجراء شدن را بغیر از cpu دارد.و هنگامیکه لازم باشد پردازش در حال اجراء برای بروز رخدادی مثلا I/O منتظر بماند،پردازش از حالتRunning بحالتBlock (انتقال 1) انتقال میابد.در اینحالت همچنان پردازش در حافظه اصلی قرار دارد و بعد از اتمام عملیات I/O بحالت Ready باز میگردد.مثلا هنگامیکه عملیات I/O پایان میابد وقفه ای از طرف I/o صادر شده و برنامه از حالتReady منتقل میشود.



بسیاری از سیستم عاملها بغیر از سه حالتReady,Running,Block حالت دیگری بنام Suspend (معلق)دارند،در حالت قبلی هنگامیکه چند برنامه در حالتReady هستند پردازنده با سرعت زیاد تمامی آنها را اجراء کرده و معمولا پردازشهای موجود درRam همگی در حالتBlock میروند در اینحالت حافظه اصلی ما پر است و پردازنده بیکار است در نتیجه برنامه جدیدی از جدول ISPT (حالتHold) نمیتواند بحالت Ready منتقل شود.برای برطرف کردن این مشکل پردازشهایی که زمان زیادی را در حالتBlock هستند به حالت معلق منتقل میشوند ،یعنی تا پایان یافتن عملیات I/O به هارد انتقال میابند،در نتیجه برنامه های جدید میتوانند از حالت Hold بهReady انتقال یابد.

Blockکنترل پردازش

PCB(Process Control Block)

پردازش، برنامه در حال اجراست ،از دیدگاه سیستم عامل میتوان گفت پردازش تشکیل شده است از یکسری ساختمان داده بنام بلاک کنترل پردازش Process Control Block(PCB)

PCB شامل اطلاعات زیادی در مورد پردازش میباشد که لازم است ذخیره شود که اگرprocess نیاز بود مجددا بحالت running برگردد و پردازش را از همان نقطه که قطع شده بود ادامه دهد.

اطلاعات درون PCB

1. حالت جاری پردازش:که میتواندReady,running.....باشد.
2. شماره شناسایی پردازش: process identifier(PID)
3. آدرس دستورالعمل بعدی:آدرسی است که خط بعدی اجراء را برای پردازش مشخص میکند.
4. محل حفظ ثباتها
5. اطلاعات زمانبندی cpu:که میتواند شامل صف های پردازشهای آماده اجراء باشد.
6. اطلاعات مدیریت حافظه:مثلا آدرس قرار گیری پردازش در حافظه
7. اطلاعات وضعیتI/O:مشخص میکند کدام پردازش چهI/O هایی را در اختیار دارد.
8. اطلاعات حسابرسی:مانند میزان زمان استفاده پردازش ازcpu

وقتی که سیستم عاملcpu را به پردازشی میدهد با استفاده ازPCB تمامی اطلاعات راه اندازی مجدد پردازش را فراهم میکند،

تعویض متن (Context Switch):به عملیات تعویض پردازشهای در حال اجراءگفته میشود .عملیات تعویض متن معمولا با سرعت زیاد انجام میگردد ولی این عملیات خود ،سرباری بر رویcpuاست.معمولا زمان تعویض متن بین 1 تا1000 میکروثانیه تغییر میکند،که مشاهده میشود این زمان آنقدر زیاد نیست که قابلیت چند برنامه گی را از بین ببرد.

به تعداد برنامه های اجرا شدهPCB وجود دارد و عملیاتcontext switch در حقیقت تعویض بینPCB هاست.

www.freebay.ir

مفهوم زمانبندی

هدف چند برنامه گی این است که در همه اوقات پردازشی در حال اجراء وجود داشته باشدتا بهره وریcpuماکزیمم شود،هدف اشتراک زمانی این است که cpu مابین پردازشها بقدری مکرر سوئیچ شودکه کاربران احساس کنند بصورت کامل با سیستم در حال محاوره هستند.

زمانی که بیش از یک پراسس وجود قابل اجراء باشد سیستم عامل باید تصمیم بگیرد کدام پراسس باید اول اجراء شود،بخشی از سیستم عامل که این تصمیم گیری را انجام میدهد زمانبند یاschedulerنامیده میشود.

انواع زمانبند

انحصاری-غیر انحصاری

انحصاری:
در سیستم انحصاری فقط هنگامی cpu از پردازش در حال اجراء گرفته میشود که یا آن پردازش نیاز به عمل I/O دارد و یا پردازش خاتمه پیدا کرده باشد.بنابراین پیاده سازی چنین الگوی زمانبندی ساده است ولی ممکن است پردازشی cpu را برای مدت طولانی در اختیار گرفته و آنرا آزاد نکند،چنین زمانبندی برای سیستمهای اشتراک زمانی مناسب نمیباشد.

غیرانحصاری:
در اکثر سیستمها از یک تایمر داخلی برای ایجاد وقفه های سخت افزاری استفاده میشود تا cpu را از پردازش در حال اجراء بگیرد و به پراسس دیگری تخصیص داده شود.چنین زمانبندی را که پس از اتمام برش زمانی cpu را از پردازشی میگیرد زمانبندی غیر انحصاری میگویند.

حد پایین برش زمانی کهcpu به هر پردازش داده میشود به دو پارامتر زیر وابسته است.

(1) هزینه ثابت تعویض متن:
به این معنی که زمان برش زمانی نباید کمتر از زمان تعویض متن باشد.

(2) زمان اجرای یک نمونه فعل و انفعال:
به این معنی که اگر برش زمانی خیلی کوتاه باشد برای اجرای هر کار کوچکی نیاز به چند برش زمانی داریم.

معیارهای زمانبندی

قبل از پرداختن به الگوریتمهای زمانبندی ،معیارهای مقایسه این الگوریتمها بررسی میگردد.

(1) عدالت: **Fairness**
اطمینان از اینکه هر پراسس یا پردازش سهم عادلانه ای ازcpu را دریافت میکند.

(2) کارائی یا بهره وریcpu:
یعنی cpu در تمام زمانها حدالمکان مشغول باشد.

(3) زمان پاسخ: **Response Time**
به حداقل رساندن زمانها یا دستورات محاوره ای

(4) زمان گردش: **Turn around**
به حداقل رساندن زمانیکه کاربر دستوری را اجراء کرده تا پاسخ کامل اجرای آن دریافت شود.

(5) توان عملیاتی یا گذردهی:
به تعداد پردازشها در واحد زمان گفته میشود.

(6) زمان انتظار: **Waiting Time**
به زمانی گفته میشود که هر پردازش منتظر اجرا می ماند.

```
C:\> dir [enter]
```

زمان پاسخ
زمان گردش

نکته مهم: در بحث زمانبندیها رعایت اولویتها بسیار مهم است.اولویتها میتوانند بصورت داینامیک یا استاتیک باشند.

در اولویت استاتیک در همان ابتدا به هر پردازش اولویت ثابتی اختصاص میابد ولی در اولویت داینامیک پردازش میتواند در زمان اجرا شدنش دارای اولویتهای مختلفی باشد ،بعنوان مثال در ابتدا اولویتش پایین است ولی در روند اجرایش اولویتش افزایش پیدا میکند.

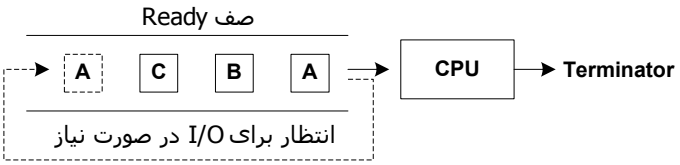
الگوریتمهای زمانبندی

در بررسی الگوریتمهای زمانبندی معیارهای گفته شده بین الگوریتمها را مقایسه میکنیم.

(1) زمانبندی F C F S: **First come First Service** انحصاری

در بررسی الگوریتمهای زمانبندی معیارهای گفته شده بین الگوریتمها را مقایسه میکنیم.

به این الگوریتم F I F O نیز گفته میشود ،این الگوریتم از نوع انحصاری میباشد
برنامه ای که cpu را در اختیار میگیرد یا باید اجرایش تمام شود و یا در صف انتظار برای I/O قرار گیرد.

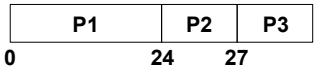


متوسط زمان انتظار برای F I F O قالبا طولانیست.

مثال 1

سه پردازش در زمان صفر در وضعیتReady هستند اگر زمان انفجار(زمان مورد نیاز برای اجراء شدن)هر پردازش با توجه به جدول زیر باشد و پردازشها بترتیب داده شده باشند برای بررسی معیار زمان انتظار میتوان از نمودار گانت **(Gannt Chart)** استفاده گرد.

پردازش کار	P1	P2	P3
زمان انفجار	24	3	3

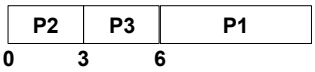


$$\text{زمان انتظار} = \frac{\text{زمان انتظار P1} + \text{زمان انتظار P2} + \text{زمان انتظار P3}}{\text{واحد 3}} = \frac{0 + 24 + 27}{3} = 17$$

مثال 2

اگر نمودار ورود پردازشها بصورت زیر باشد نمودار گانت آنرا گشیده و میانگین زمان انتظار را محاسبه کنید.

پردازش کار	P2	P3	P1
زمان انفجار	3	3	24



$$\text{میانگین زمان انتظار} = \frac{0+3+6}{3} = 3$$

مزایا: **FIFO**

- سادگی پیاده سازی و عدالت نسبی

معایب: **FIFO**

- سرعت کم

روش **FIFO** معمولا باعث اثر اسکورت میشود ،بعنوان مثال وقتی برای تفریح به شمال میرویم یک کامیون در جلو و بقیه خودروها پشت سر آن اسکورت میشوند.

Bay

www.freebay.ir

(2) روش نوبت گردش: **Round Robin(RR)** غیرانحصاری

این زمانبندی از قدیمیترین و ساده ترین و عدلانه ترین و رایج ترین الگوریتمهای زمانبندی میباشد.این روش غیر انحصاری است.بعبارت دیگر یک واحد زمانی کوچک بنام برش زمانیTime slice به هر پردازش اختصاص داده میشود،و میانگین زمان انتظار الگوریتمRR غالبا طولانیست.

مثال 1

اگر پردازشها در زمان صفر وارد شده باشند و برش زمانی4 واحد زمانی باشد نمودار گانت آنرا کشیده و میانگین زمان انتظار را برای آن مشخص کنید.

پردازش کار	P1	P2	P3
زمان انفجار	24	3	3

20

P1	P2	P3	P1	P1	P1	P1	P1

0 4 7 10 14 18 22 26 30

اولین زمان انتظار **P1=0**

دومین زمان انتظار **P1=10- 4=6**

میانگین زمان انتظار $= \frac{(0+6)+4+7}{3} = 5.4$

مثال 2

اگر پردازشها در زمان صفر وارد شده باشند و برش زمانی 1 واحد زمانی باشد نمودار گانت آنرا کشیده و میانگین زمان انتظار را برای آن مشخص کنید.

پردازش کار	P1	P2	P3
زمان انفجار	24	3	3

21

P1	P2	P3	P1	P2	P3	P1	P2	P3	P1

0 1 2 3 4 5 6 7 8 9 30

میانگین زمان انتظار $= \frac{(0+2+2+2)+(1+2+2)+(2+2+2)}{3} = 5.66$

(3) زمانبندی کوتاهترین کار: **Shortest Job First (SJF)** انحصاری

این الگوریتم روشی انحصاری است کهcpu به پردازشی داده میشود.که کوتاهترین زمان انفجار محاسباتی بعدی را داشته باشد.نکته مهم اینست که زمانبندی بر اساس طول مدت انفجار محاسباتی بعدی عمل میکند و نه بر اساس زمان محاسباتی کل پردازش

اگر چند برنامه دارای انفجار محاسباتی یکسانی باشند زمانبندی را برای آن برنامه ها میتوان بصورتFCFS(FIFO) انجام داد.

با توجه به جدول زیر نمودار گانت و همچنین میانگین زمان انتظار را با توجه به الگوریتم SJFو FIFO محاسبه کنید؟

پردازش کار	P1	P2	P3	P4
زمان انفجار	6	8	7	3

(FIFO)

P1	P2	P3	P4

0 6 14 21

$\frac{0+6+14+21}{4} = 10.25$

(SJF)

P4	P1	P3	P2

0 3 9 16

$\frac{0+3+9+16}{4} = 7$

همانطور که مشاهده میشود روشSJF مناسبتر از روش FIFO میباشد ،میتوان ثابت کرد روشSJF از نظر میانگین انتظار بهینه ترین روش زمانبندی است ،این الگوریتم در برنامه های دسته ای که از قبل زمان اجرای آن مشخص است مناسب میباشد ،مهمترین مشکل این روش آن است که آگاهی دقیقی از طول زمان انفجار محاسباتی بعدی نداریم ،بنابراین میتوانیم از روش حدس زدن استفاده کنیم به اینصورت که طول انفجار محاسباتی بعدی خیلی شبیه طول انفجار محاسباتی قبلی باشد.

مشکل دیگر قحطی زدگی(Starvation) میباشد ،به این مفهوم که همواره در سیستم به پردازشهای کوچک سرویس داده میشود و پردازشهای بزرگ در انتهای صف قرار میگیرند ،با ورود پردازشهای جدید کوچک همواره پردازشهای بزرگ در انتهای صف در وضعیت قحطی زدگی قرار میگیرند.

با توجه به ایکه این روش از نوع انحصاری میباشد از اینرو مناسب کاربردهای محاوره ای نیست.(Time sharing)

(4) روش کوتاهترین زمان باقیمانده:**SRT (Shortest Remaining Time)** غیرانحصاری

در این روش مانند SJF زمانبندی صورت میپذیرد ولی از نوع غیر انحصاری ،درSRT برنامه ای که احتیاج به کمترین زمان جهت تکمیل شدن دارد ابتدا اجرا میشود و در هنگام انتخاب یک پردازش ،پردازشهایی را که تازه به صف آمده اند نیز در نظر گرفته میشود،در این حالت ممکن است cpu از برنامه ای که در حال اجراست و زمان تکمیل بیشتری نسبت به برنامه موجود در صف دارد گرفته شود.(روش غیر انحصاری)

در این روش نیز مشکل محاسبه تخمین آینده را داریم و همچنین احتمال بروز قحطی زدگی نیز وجود دارد.

با توجه به جدول زیر نمودار گانت و همچنین میانگین زمان انتظار را با توجه به الگوریتم SRT محاسبه کنید؟

پردازش کار	P1	P2	P3	P4
زمان انفجار	8	4	9	4
زمان ورود	0	1	2	3

P1	P2	P4	P1	P3

0 1 5 9 16

SRT

میانگین زمان انتظار $= \frac{P1 \quad P2 \quad P3 \quad P4}{\frac{[0+(9-1)]+[1-1]+[16-2]+[5-3]}{4}} = 6$

P1	P2	P4	P3

0 8 12 16

SJF

میانگین زمان انتظار $= \frac{P1 \quad P2 \quad P3 \quad P4}{\frac{[0]+[8-1]+[16-2]+[12-3]}{4}} = 7.5$

پردازشهای جدول زیر را با برش زمانی1 به روشRR پیاده کنید و همچنین نمودار گانت آن و میانگین زمان انتظار را برای آن تعیین کنید.

پردازش کار	P1	P2	P3	P4
زمان انفجار	8	4	9	4
زمان ورود	0	2	4	5

P1	P1	P1	P2	P1	P2	P3	P1	P4	P2	P3	P1

0 1 2 3 4 5 6 7 8 9 10 11 12

P1 8 P1 7 P1 6 P2 4 P1 5 P2 3 P3 9 P1 4 P4 4 P2 2 P3 8 P1 3 P4 3

P2 4 P1 5 P2 3 P3 9 P1 4 P4 4 P2 2 P3 8 P1 3 P4 3 P2 1

P3 9 P1 4 P4 4 P2 2 P3 8 P1 3 P4 3 P2 1 P3 7

P4 4 P2 2 P3 8 P1 3 P4 3 P2 1 P3 7 P1 2

P4	P2	P3	P1	P4	P3	P1	P4	P3	P3	P3	P3	P3

13 14 15 16 17 18 19 20 21 22 23 24

P2 1 P3 7 P1 2 P4 2 P3 6 P1 1 P4 1 P3 5

P3 7 P1 2 P4 2 P3 6 P1 1 P4 1 P3 5

P1 2 P4 2 P3 6 P1 1 P4 1 P3 5

P4 2

P1	(0-0)+1+2+3+3+2	$\frac{P1+P2+P3+P4}{4} = 10.5$
P2	(3-2)+1+3+3	
P3	(6- 4)+3+3+2+2	
P4	(8- 5)+3+3+2	

Bay

www.freebay.ir

5 روش بالاترین نسبت پاسخ: **HRRN Highest Response Ratio Next** انحصاری

این زمانبندی از نوع انحصاری میباشد که برخی از مشکلات SJF را برطرف نموده است ،مشکل اصلی SJF قحطی زدگی بود که در روش فوق با اختصاص اولویت Dynamic مشکل قحطی زدگی را با استفاده از فرمول زیر برطرف میکنیم .

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

بعلت اینکه زمان سرویس در مخرج است بنابراین کارهای کوتاهتر اولویت بیشتری پیدا میکنند و زودتر اجرا میشوند ،ازطرف دیگر چون پارامتر زمان انتظار در صورت است کارهای طولانی تر که مدت زیادی را در صف گذرانده اند اولویتشان افزایش میابد ،بنابراین مشکل قحطی زدگی برطرف میگردد

6 روش زمانبندی اولویت: **Priority Scheduling** غیرانحصاری-انحصاری

این روش میتواند به دو صورت انحصاری و غیر انحصاری پیاده سازی شود ولی معمولا بصورت انحصاری پیاده میشود. در بسیاری از محیطها بحث اولویت مطرح است به این مفهوم که پردازشهایی که دارای اولویت بالاتری هستند در ابتدا سرویس دهی میشوند و بعد از سرویس دهی به این پردازشها به پردازشهایی با اولویت کمتر سرویس داده میشود ،معمولا در پیاده سازی اولویت،عدد کوچکتر اولویت بیشتر دارد.

در مثال زیر پردازشها همگی در زمان صفر وارد شده اند نمودار گانت و میانگین زمان انتظار را برای آن محاسبه کنید.

پردازش کار	P1	P2	P3	P4	P5
زمان انفجار	10	1	2	1	5
اولویت	3	1	3	4	2

P2	P5	P1	P3	P4

0 1 6 16 18 19

$$\frac{0+1+6+16+18}{5} = 8.2$$

در مثال زیر پردازشها در زمان مختلف وارد شده اند نمودار گانت و میانگین زمان انتظار را برای آن محاسبه کنید.(انحصاری)

پردازش کار	P1	P2	P3	P4	P5
زمان انفجار	10	1	2	1	5
اولویت	3	1	3	4	2
زمان ورود	0	9	11	12	14

P1	P2	P3	P4	P5

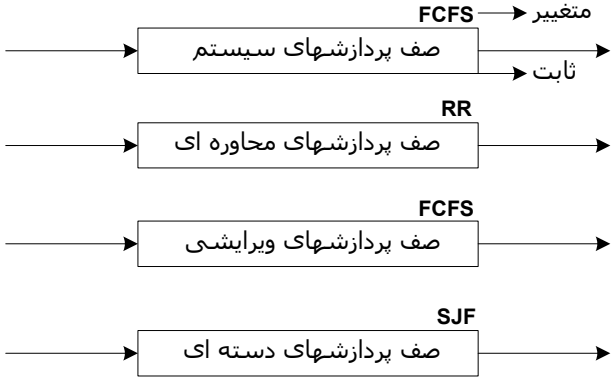
0 10 11 13 14 19

زمان انتظار **P1=0**
زمان انتظار **P2=(10-9)=1**
زمان انتظار **P3=(11-11)=0**
زمان انتظار **P4=(13-12)=1**
زمان انتظار **P5=(14-14)=0**

$$\frac{2}{5} = 0.4$$

7 روش زمانبندی صفهای چند گانه: **Multiple Queue**

هنگامیکه بتوان پردازشی را به دسته های متفاوتی تقسیم بندی کرد از این روش استفاده میکنیم ،مثلا ممکن است پردازشها به چند دسته بصورت زیر تقسیم بندی گردد.اولویت صفها با هم فرق میکند هر صف بصورت جداگانه ای میتواند الگوریتم زمانبندی جداگانه ای داشته باشد ،این الگوریتم از نوع انحصاری میباشد و تا زمانیکه صف با اولویت بالاتر تمام نشده است صفهای با اولویت پایینتر شروع نمیشوند.



8 زمانبندی صفهای چند گانه با فیدبک: **Multilevel Feedback Queue (MFQ)**

در روش قبلی یک فرایند وقتی وارد صفی میشد تا کامل شدن اجرایش نمیتوانست از آن صف خارج شود و وارد صف دیگری گردد،ولی در روش MFQ یک پردازش میتواند بین صفها جابجا شود به این صورت که یک پردازش در هنگام ورود در صفی با اولویت پایینتر وارد میشود و هنگامیکه فرایندها در مدت زمان زیادی در صف با اولویت پایین قرار میگیرند میتوانند به صفی با اولویت بالا انتقال یابند در این روش نیز یک صف باید کاملا اجرا گردد تا صف دیگری اجراء گردد. ساختار کلی آن بصورت زیر میباشد.

کوانتوم = برش زمانی

9 زمانبندی شانسی:

10 زمانبندی بزرگترین زمان درخواست :

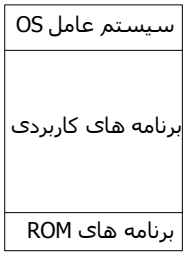
یک روش انحصاری است وبرعکس روش SJF عمل میکند بدین صورت که به بزرگترین پردازش موجود در رم CPU اختصاص داده میشود.

مدیریت حافظه

بخشی از سیستم عامل است که سلسله مراتب حافظه را اداره میکند ،این بخش باید بداند که کدام قسمت حافظه خالی و کدام قسمت استفاده شده است و هنگامیکه چندین برنامه بصورت همزمان در حافظه قرار دارد،وظیفه مدیر حافظه کنترل جلوگیری از تداخل این برنامه ها میباشد.

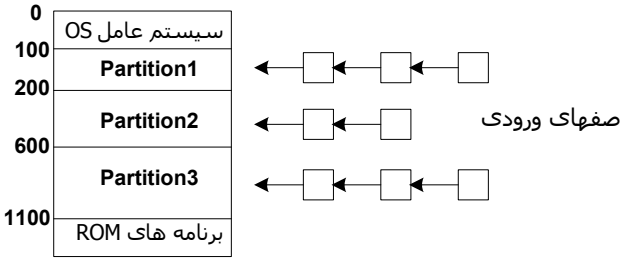
روشهای مدیریت حافظه

1 تک برنامه ای ساده: **Single Partition Allocation**



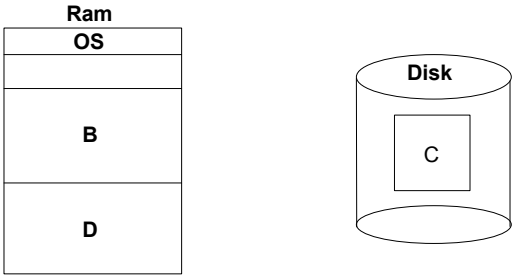
در این مدل در قسمت برنامه کاربردی تنها یک برنامه قابل اجراء است ،بدیهیست که اگر برنامه مورد نظر به میزان فضای بیشتری از فضای موجود در ram نیاز داشته باشد آن برنامه در چنین سیستمی قابل پیاده سازی نمیباشد.

2 چند برنامه گی با بخشبندی ثابت حافظه: **Multi/Static Partition Allocation**



در این روش فضای رم به چندین قسمت ثابت تقسیم میشود که ممکن است این قسمتها اندازه های مختلفی داشته باشند ،هر یک از این قسمتها میتواند پذیرای یک پردازش جداگانه باشد در نتیجه امکان اجرای چند برنامه گی میسر است .مشکل بزرگ این روش اینست که اگر برنامه ای داشته باشیم که برای اجراء شدن نیاز به حافظه ای به اندازه رم داشته باشد ،با این روش امکان اجرای آن پردازش در سیستم مورد نظر وجود ندارد.

Bay
www.freebay.ir



تبدیل آدرسها

در اکثر سیستمها برنامه ها میتوانند در فضاهای متفاوتی در حافظه قرار گیرند ،از اینرو پیوندهای آدرسهای حافظه میتواند به یکی از سه صورت زیر انجام شود.

1- تبدیل آدرس در زمان کامپایل:
در این روش برنامه ها در موقع کامپایل شدن مشخص میشود که در چه مکانی از رم قرار میگیرند.مانند برنامه های با پسوند com

2-تبدیل آدرس در زمان بارگذاری:
در این روش برنامه ها هنگامیکه برای اولین بار وارد رم میشوند آدرسشان تغییر میکند.

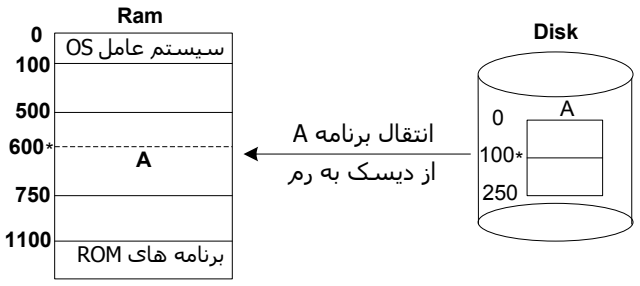
3- تبدیل آدرس در زمان اجراء:
اگر پردازشی بتواند در زمان اجرایش در حافظه رم جابجا شود آنگاه تبدیل آدرس تا هنگام اجرا به تاخیر می افتد.

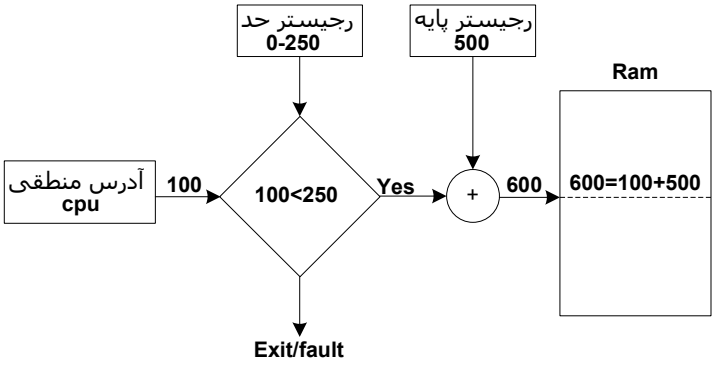
آدرس فیزیکی و آدرس منطقی

آدرس منطقی:
آدرسی که در برنامه قرار دارد و CPU با آن آدرس سروکار دارد آدرس منطقی نامیده میشود.

آدرس فیزیکی:
آدرسی است که در مکان فیزیکی رم به یک برنامه اختصاص داده میشود.

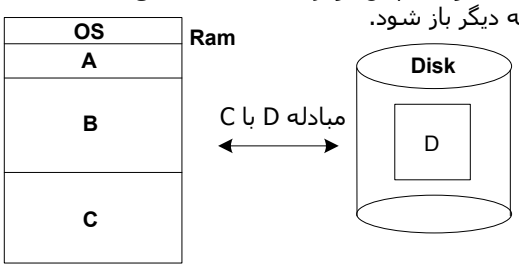
شکل زیر عملیاتی که cpu برای تبدیل آدرس منطقی به آدرس فیزیکی را انجام میدهد مشخص میکند که در این شکل از دو ثبات (رجیستر) حد و پایه استفاده میشود،





3 روش مبادله: **Swapping**

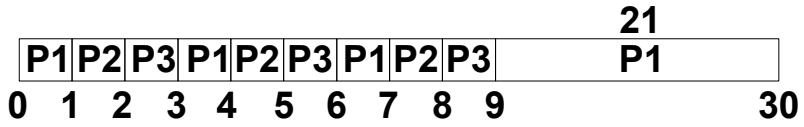
در این روش به اینصورت عمل میشود که برخی مواقع حافظه رم به اندازه ای نیست که بتولند تمامی برنامه های در حال اجراء را در خود جای دهد ،در سیستم مبادله هر پردازش بصورت کامل به حافظه آورده میشود و برای مدتی اجراء میشود ،سپس دوباره به دیسک منتقل شده تا فضا برای اجرای برنامه دیگر باز شود.



Bay
www.freebay.ir

اگر پردازشها در زمان صفر وارد شده باشند و برش زمانی 1 واحد زمانی باشد نمودار گانت آنرا کشیده و میانگین زمان انتظار را برای آن مشخص کنید.

پردازش کار	P1	P2	P3
زمان انفجار	24	3	3

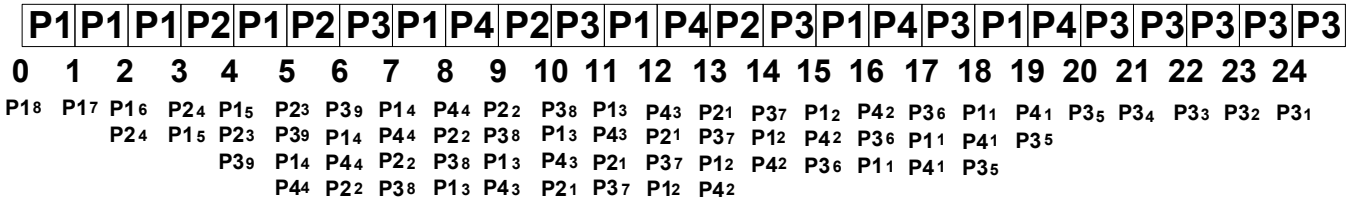


$$\text{میانگین زمان انتظار} = \frac{(0+2+2+2)+(1+2+2)+(2+2+2)}{3} = 5.66$$

www.freebay.ir

پردازشهای جدول زیر را با برش زمانی 1 به روش RR پیاده کنید و همچنین نمودار گانت آن و میانگین زمان انتظار را برای آن تعیین کنید؟

پردازش کار	P1	P2	P3	P4
زمان انفجار	8	4	9	4
زمان ورود	0	2	4	5



زمان انتظار **P1** **0+1+2+3+3+2=11**

زمان انتظار **P2** **1+1+3+3=8**

زمان انتظار **P3** **2+3+3+2+2=12**

زمان انتظار **P4** **3+3+3+2=11**

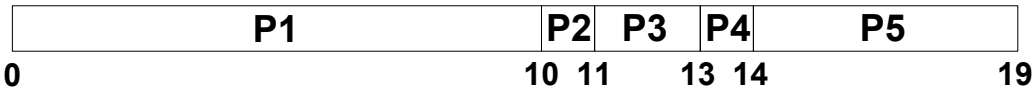
$$\text{میانگین زمان انتظار} = \frac{11+8+12+11}{4} = 10.5$$

بابک آشفته یزدی-پودمان 4 –دانشگاه علمی کاربردی شهر قدس

www.freebay.ir

در مثال زیر پردازشها در زمانهای مختلف وارد شده اند نمودار گانت و میانگین زمان انتظار را برای آن محاسبه کنید.(انحصاری)

پردازش کار	P1	P2	P3	P4	P5
زمان انفجار	10	1	2	1	5
اولویت	3	1	3	4	2
زمان ورود	0	9	11	12	14



زمان انتظار **P1** **=0**

زمان انتظار **P2** **(10-9)=1**

زمان انتظار **P3** **(11-11)=0**

زمان انتظار **P4** **(13-12)=1**

زمان انتظار **P5** **(14-14)=0**

$$\text{میانگین زمان انتظار} = \frac{0+1+0+1+0}{5} = 0.4$$

www.freebay.ir