

پلسه ی دو ۴:

عموما کارها در دو دسته قرار می گیرند

- 1- کارهای I/O bound (I/O Limited): کارهایی که بخش زیادی از اجرای آنها در ارتباط با دستگاههای ورودی/خروجی بوده و مسابسات زیادی ندارند. مثل برنامه ای که می بایست کارنامه ی دانشجویان را چاپ کند
  - 2- کارهای CPU bound (CPU Limited): کارهایی که مهم زیادی مسابسات داشته و بخش عمده نیاز آنها برای اجرا وقت پردازنده است. مثل برنامه ای که می بایست یک دستگاه معادله ی صد مجهولی را حل کند
- نکته:

در مورد گاهای CPU bound و I/O bound استفاده از بافر تاثیر زیادی در بهبود اجرا ندارد.

زمانبند بلند مدت می بایست تلفیق مناسبی از کارهای CPU bound و I/O bound را انتخاب کند.

حالت پایدار: زمانی که نرخ ایجاد پردازها برابر با نرخ خروج یا فائمه پردازها باشد. بنا بر این زمانبند بلند مدت سعی می کند که سیستم به حالت پایدار برسد.

الگوریتم های زمانبندی کوتاه مدت (پردازنده):

عمل زمانبندی کوتاه مدت بر اساس الگوریتم های زمانبندی انجام می شود. این الگوریتم ها سعی در برآورده کردن معیارهای زیر را دارند.

1- عدالت: یعنی سهم پردازها از CPU به طور منصفانه باشد

2- افزایش کارایی: هدف از کارایی مشغول بودن CPU به طور ایده آل است.

3- افزایش گذردهی: منظور تعداد پردازها می باشد که در واحد معینی از زمان می باشد.

4- کاهش زمان پاسخ: زمان ما بین مطرح شدن یک پردازها تا اولین اجرای آن پردازها می باشد.

این اهداف کاملا باهم در تناقض اند بنا براین می بایست سعی شود مصالحه ای بین این اهداف صورت گیرد.

زمان انتظار (waiting Time): مدت زمانی را که پردازها در حالت آماده است و منتظر CPU می باشد را زمان انتظار گویند.

زمان اجرا (Runing Time): مدت زمانی را که پردازها بر روی CPU در حال اجراست را زمان اجرا گویند

زمان I/O: مدت زمانی است که پردازها نیاز به عمل I/O دارد

زمان گردش کار (Turnatound Time): مدت زمان ما بین ورود یک پردازها به سیستم تا اجرای کامل یک پردازها را زمان گردش کار گویند.

زمان تعویض متن + زمان I/O + زمان اجرا + زمان انتظار = زمان گردش کار

زمان تعویض متن: زمانی است که برای تعویض پردازها بر روی CPU صرف میشود. که این عمل شامل زفیره حالت پردازها در

حال اجرا Update ثبات های سیستم به مقادیر پردازها جدید می باشد.

الگوریتم های زمانبندی به دو دسته تقسیم می شوند.

1- انحصاری یا انقطاع نا پذیر (Preemptive): در این الگوریتم ها تا زمان فائمه پردازها و یا تا زمان نیاز به عمل I/O نمی توان CPU را از

پردازها در حال اجرا گرفت و به پردازها دیگری انتساب داد.

2- غیر انحصاری یا انقطاع پذیر (Non Preemptive): در این الگوریتم ها در پایان برش زمانی (Time Slice) و یا تغییر شرایط سیستم، مدیر زمانبندی می تواند کنترل پردازنده را از یک پردازش در حال اجرا گرفته و به پردازش دیگری بدهد.

الگوریتم الویت با اولین ورودی -FCFS- (Frist In Frist Out یا Frist Come Frist Service)

این الگوریتم به سادگی هر پردازش ورودی را در صف FIFO قرار داده و از سر صف اجرائش را شروع می کند و اجرای پردازش ها را به طور انحصاری انجام میدهد.

مزایا:

سادگی اجرا- عملی بودن

معایب:

1- زیاد بودن میانگین زمان انتظار (زمان گردش کار)

2- انقطاع نا پذیری و غیر قابل استفاده بودن در سیستم های اشتراک زمانی است

الگوریتم الویت با کوتاهترین کار- SJF (Shortest Job Frist):

این الگوریتم از بین پردازش های موجود در صف آماده، پردازش ای را جهت اجرا انتخاب می کند که زمان اجرای کمتری نیاز داشته باشد و پردازش در حال اجرا اجرائش را به صورت انقطاع ناپذیر ادامه می دهد.

معایب:

1- نیاز به داشتن اطلاعات مقاطع زمانی مورد نیاز قبل از شروع اجرا

2- احتمال به تعویق افتادن کارهای طولانی

3- انقطاع نا پذیری

مزایا: دارا بودن کمترین زمان برگشت یا زمان انتظار ما بین تمام الگوریتم ها

الگوریتم الویت با کمترین زمان باقیمانده -SRT (Shortest Remaining Time)

در این الگوریتم انتخاب بر اساس کمترین زمان مورد نیاز برای کامل شدن صورت میگیرد. این الگوریتم غیر انحصاری بوده و با ورود هر پردازش به صف آماده، بر رسی زمان باقیمانده پردازش ها انجام می گیرد، اگر پردازش تازه وارد شده، زمان کمتری برای کامل شدن لازم دارد، پردازنده در اختیار آن قرار می گیرد.

معایب: احتمال تعویق کارهای طولانی- نیاز به دانستن زمان مورد نیاز پردازش ها

مزایا: انقطاع پذیری- زمان پاسخ نسبتا مناسب

مثال: در سیستمی سه پردازش  $P_0$  و  $P_1$  و  $P_2$  مطابق جدول زیر با زمان های ورود و اجرای داده شده قرار دارند مطلوب است مناسبه

میانگین زمان انتظار و میانگین زمان گردش کار در حالت های زیر

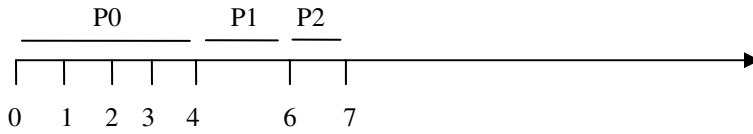
الف: برای زمانبندی پردازش ها از الگوریتم FCFS استفاده شود.

ب: برای زمانبندی پردازش ها از الگوریتم SJF استفاده شود.

ج: برای زمانبندی پردازش ها از الگوریتم SRT استفاده شود.

پردازش	زمان رسیدن	زمان اجرا
$P_0$	t	4
$P_1$	t+1	2
$P_2$	t+2	1

الف:



$$\text{میانگین زمان انتظار} = \frac{\text{مجموع زمان انتظار تمام پردازش ها}}{\text{تعداد پردازش ها}} \quad \text{میانگین زمان گردش کار} = \frac{\text{زمان اجرای پردازش + زمان انتظار}}{\text{تعداد پردازش ها}}$$

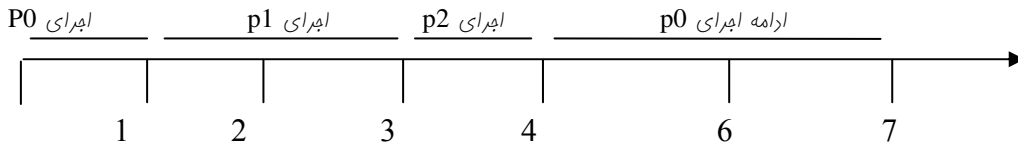
$$\text{میانگین زمان انتظار} = \frac{0+3+4}{3} = \frac{7}{3} \quad \text{میانگین زمان گردش کار} = \frac{(0+4) + (3+2) + (4+1)}{3} = \frac{14}{3}$$

ب) در این شرایط اگر پردازش ای شروع به اجرا شود تا انتها ادامه می یابد در مرحله ی بعد پردازش ای برای اجرا انتخاب می شود که کمترین زمان اجرا را دارد بنا براین اول P0 (در این لحظه فقط همین را داریم) تا لحظه ی 4 اجرا میشود در این لحظه دو تا پردازش آماده داریم که P2 زمان کمتری را برای اجرا می فواید پس تا لحظه ی 5 اجرا می شود از 5 تا 7 زمان اجرای P1 می باشد.

$$\text{زمان انتظار} : P_0=0s, P_1=4s, P_2=2s$$

$$\text{میانگین زمان انتظار} = \frac{0+4+2}{3} = 2 \quad \text{میانگین زمان گردش کار} = \frac{(0+4) + (4+2) + (2+1)}{3} = \frac{13}{3}$$

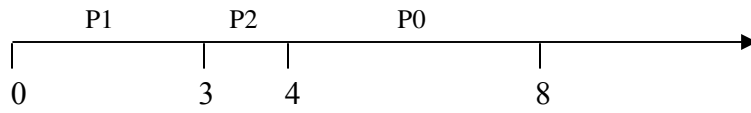
ج) در این شرایط در تمامی لحظه های اجرای پردازش ای از راه برسد که زمان اجرائیش کمتر از زمان باقیمانده پردازش در حال اجرا باشد پردازنده را در اختیار گرفته و شروع به اجرا می کند  $P_0=3s, p_1=0s, p_2=1s$  زمان انتظار



$$\text{میانگین زمان انتظار} = \frac{3+0+1}{3} = \frac{4}{3} \quad \text{میانگین زمان گردش کار} = \frac{(3+4) + (0+2) + (1+1)}{3} = \frac{11}{3}$$

مثال: مثال قبل را با جدول زیر حل کنید (برای قسمت ب و ج)

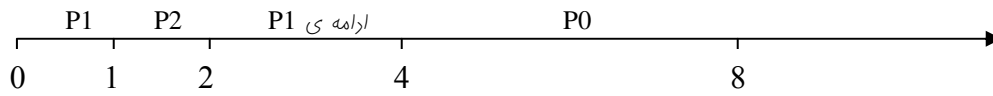
پردازش	زمان رسیدن	زمان اجرا
$P_0$	0	4
$P_1$	0	3
$P_2$	1	1



ب) SJF

میانگین زمان گردش  $ATT = \frac{(4+4) + (0+3) + (2+1)}{3} = \frac{14}{3}$

میانگین زمان انتظار  $AWT = \frac{4+0+2}{3} = 2$



ج) SRT

$ATT = \frac{(4+4) + (1+3) + (0+1)}{3} = \frac{13}{3}$

$AWT = \frac{4+1+0}{3} = \frac{5}{3}$

پایان جلسه دوم