



آزمایشگاه سیستم عامل

جلسه دوازدهم

نیمسال اول ۱۴۰۱-۱۴۰۲

الگوریتم‌های زمان‌بندی

در حالت کلی، فرآیند یک برنامه در حال اجرا است که وارد سیستم شده، روی آن پردازش انجام می‌شود و در نهایت از سیستم خارج می‌گردد اما یک فرآیند بلافاصله بعد از ورود نمی‌تواند CPU را در اختیار گرفته و عملیات پردازش روی آن صورت گیرد. الگوریتم‌های زمان‌بندی فرآیندها را بر روی پردازنده به شیوه‌ای کارآمد و مؤثر برنامه‌ریزی می‌کنند. این زمان‌بندی توسط یک زمان‌بند فرآیند انجام می‌شود که با افزایش توان عملیاتی، استفاده از CPU را به حداکثر می‌رساند.

به عبارت دیگر الگوریتم‌های زمان‌بندی پردازنده، مدیریت فرآیندها در سیستم عامل را بر عهده دارند و برای توزیع منابع بین طرف‌هایی که به‌طور هم‌زمان و غیر هم‌زمان آنها را درخواست می‌کنند، استفاده می‌شوند.

اهداف الگوریتم‌های زمان‌بندی در سیستم عامل

- حداکثر استفاده از CPU: CPU را تا حد امکان مشغول نگه می‌دارند.
- تخصیص منصفانه CPU: اختصاص عادلانه CPU به فرآیندها را انجام می‌دهد.
- حداکثر توان عملیاتی: تعداد فرآیندهایی که اجرای خود را در واحد زمان کامل می‌کنند.
- حداقل زمان بازگشت: زمان صرف شده توسط یک فرآیند برای پایان اجرا می‌باشد.
- حداقل زمان انتظار: زمان انتظار یک فرآیند در صف آماده است.
- حداقل زمان پاسخ: زمانی که یک فرآیند اولین پاسخ را تولید می‌کند.

انواع الگوریتم‌های زمان‌بندی

الگوریتم‌های زمان‌بندی پردازنده در مدیریت فرآیندهای سیستم‌عامل به دو گروه اصلی تقسیم می‌شوند.

۱. الگوریتم‌های انحصاری

۲. الگوریتم‌های غیر انحصاری

الگوریتم‌های انحصاری (Non Preemptive)

در این الگوریتم‌ها به محض اینکه یک فرآیند پردازنده را در اختیار گرفت و شروع به اجرا شدن کرد، تا زمانی که فرآیند به طور کامل به پایان نرسد و یا مسدود نشود پردازنده را در اختیار فرآیند دیگری قرار نمی‌دهد. به عبارت دیگر در این نوع از الگوریتم‌ها فرآیندها یکجا و در یک‌بار اجرا شده و به قسمت‌های کوچک‌تر تقسیم نمی‌شوند. الگوریتم‌های FCFS, SJN از این نوع هستند.

الگوریتم‌های غیر انحصاری (Preemptive)

در این الگوریتم‌ها یک فرآیند که در حال اجراست ممکن است توسط سیستم‌عامل متوقف شده و به حالت آماده منتقل شود. این کار برای اختصاص پردازنده به یک فرآیند دیگر و یا انجام عملیات I/O و وقفه انجام می‌شود. به عبارت دیگر در این نوع از الگوریتم‌ها، فرآیندها ممکن است به چند بخش تقسیم شده و در چند مرحله عملیات تخصیص پردازنده انجام گرفته و اجرا شود. الگوریتم RR از این نوع است.



P3 = 4





Shortest Job Next (SJN)

Process	Duration	Oder
P1	6	1
P2	8	2
P3	7	3



P1 Waiting Time : 0

P3 Waiting Time : 6

P2 Waiting Time : 13

The Average Waiting Time : $(0+6+13)/3=6/33$



Priority Scheduling

Process	Duration	Priority
P1	5	2
P2	8	1
P3	7	3

P2 = 8

P1 = 5

P3 = 7



P2 Waiting Time : 0

P1 Waiting Time : 8

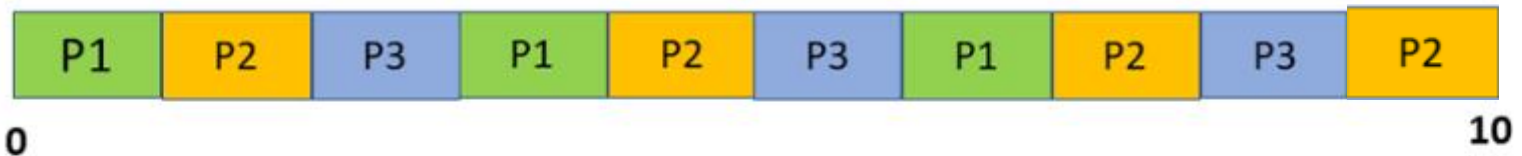
P3 Waiting Time : 13

The Average Waiting Time : $(0+8+13)/3 = 7$



Round Robin(RR)

Process	Duration	Order
P1	3	1
P2	4	2
P3	3	3



P1 Waiting Time : 7
P2 Waiting Time : 10
P3 Waiting Time : 9

The Average Waiting Time : $(7+10+9) / 3$



برای فرآیند می‌توانید از ساختاری مشابه ساختار زیر استفاده کنید:

```
struct process
{
    int pid;
    int st, wt, tt; //service time, waiting time, total time
}p[10];
```


تمرین

- برنامه‌ای به زبان C بنویسید که الگوریتم FCFS را پیاده‌سازی کند. برای این کار مراحل زیر را طی کنید:
- ۱- تعداد فرآیندها را از کاربر دریافت کنید.
 - ۲- زمان سرویس‌دهی هر فرآیند را از کاربر دریافت کنید.
 - ۳- زمان انتظار برای فرآیند اول را با صفر مقداردهی کنید.
 - ۴- زمان انتظار سایر فرآیندها را مشخص کنید. (دقت کنید که زمان انتظار یک فرآیند برابر با زمان اجرای فرآیند قبل است. زمان اجرای یک فرآیند برابر است با مجموع زمان انتظار و زمان سرویس‌دهی).
 - ۵- زمان اجرای فرآیندها را حساب کنید.
 - ۶- متوسط زمان انتظار و زمان اجرا برای هر فرآیند را حساب کنید و نمایش دهید.