



دانشگاه اصفهان
دانشکده علوم ریاضی و کامپیوتر

Operating Systems

اصول سیستم‌های عامل

سری سوم تمرینات
نیم‌سال اول ۱۴۰۳-۱۴۰۴

اعضای گروه

محمد ملائی : ۴۰۱۴۰۲۳۰۴۲

علیرضا احمدی وشکی : ۴۰۱۴۰۱۳۰۰۷

یگانه رستگاری : ۴۰۱۴۰۱۳۰۴۰

نام استاد درس

مجتبی رفیعی

تمرین ۱

صحیح یا غلط بودن گزاره‌های زیر را مشخص کنید و دلیل خود را نیز بیان کنید.

(آ) انتخاب فرآیند برای اجرا توسط زمان بند CPU انجام می شود. گزاره صحیح است. این مولفه از میان فرآیندهای آماده‌ی اجرا در حافظه، فرآیندی را انتخاب و CPU را به آن اختصاص می دهد.

(ب) طرح های غیرقبضه ای در حالت کلی سربار بیشتری را به سیستم تحمیل می کنند. گزاره غلط است. طرح های قبضه ای به طور کلی سربار بیشتری به سیستم تحمیل می کنند اما خدمات بهتری به کاربر ارائه می کنند. یکی از دلایل آن تعویض متن غیرداوطلبانه است. ممکن است در حین اجرای فرایند، به علت ورود فرایند با اولویت بالاتر یا تمام شدن کوانتوم زمانی، CPU از فرایند گرفته شود و به فرایند دیگری داده شود و تعویض متن داریم. درحالی که در طرح غیرقبضه ای، تعویض متن در صورتی انجام می شود که اجرای فرایند به اتمام برسد یا وقفه ای رخ دهد. به طور کلی، در طرح غیرقبضه ای تعویض متن کمتر و در نتیجه سربار کمتری داریم.

(ج) زمان اجرای کامل یک فرآیند برابر حاصل جمع زمان پاسخ و زمان اجرا است. به طور کلی، گزاره غلط است زیرا با تعریف اصلی زمان اجرای کامل همخوانی ندارد. زمان اجرای کامل (زمان بازگشت) فرایند، برابر است با مجموع زمان انتظار و زمان اجرای فرایند. زمان پاسخ یعنی فاصله زمانی ورود فرایند به صف تا اولین باری که CPU را دریافت می کند. در طرح زمان بندی غیرقبضه ای این گزاره صحیح است چرا که زمان پاسخ با زمان انتظار فرایند برابر است. اما به طور کلی آن را غلط محسوب می کنیم.

(د) در تابع انتخاب زمان بندی RR فرآیندی انتخاب می شود که از آخرین باری که به صف آماده اضافه شده است تا لحظه ی کنونی، زمان کمتری در صف آماده منتظر بوده است. گزاره غلط است. در این تابع، پس از هر کوانتوم زمانی فرایندی که در ابتدای صف باشد انتخاب می شود. ممکن است در همان لحظه فرایندی به صف اضافه شود و طبق این گزاره، باید توسط الگوریتم انتخاب شود زیرا کمترین زمان را در صف سپری کرده. اما این اتفاق نمی افتد. فرایند به انتهای صف اضافه شده و الگوریتم فرایندی را که در ابتدای صف است انتخاب می کند.

(ه) نحوه ی انتظار برای دریافت مجوز ورود به ناحیه ی بحرانی توسط یک فرآیند تنها با وقفه صورت می گیرد. گزاره غلط است. دو حالت کلی برای دریافت مجوز و ورود به فرایندها به ناحیه بحرانی وجود دارد: انتظار مشغول (busy waiting) : فرآیند مکرراً شرطی را بررسی می کند و صبر می کند تا شرط برقرار شود و به ادامه اجرای خود پردازد. مسدود کردن (blocking) : اجرای فرآیند به طور کامل متوقف می شود.

تمرین ۲

به سوالات زیر پاسخ کامل دهید.

(آ) در الگوریتم دکر (تلاش سوم) با استفاده از چه تغییراتی نسبت به تلاش دوم ویژگی انحصار متقابل رعایت شد؟ در تلاش دوم داشتیم که هر فرآیند ابتدا شرایط فرآیند دیگر را چک می‌کند و بعد flag خودش را TRUE می‌کند. ایده این است: فرآیند ابتدا بررسی میکند که آیا اجازه‌ی ورود به حلقه برای اجرای «ناحیه بحران» را دارد یا خیر. اگر این اجازه را داشت به فرآیند دیگری اجازه‌ی ورود به ناحیه بحرانی را نمی‌دهد.

مشکل این روش آنجاست که در زمانبندی قبضه‌ای هر لحظه امکان قبضه شده یک فرآیند و دادن پردازنده به فرآیند دیگری وجود دارد. بنابراین منطقی است که هر فرآیند ابتدا مطمئن شود که فرآیند دیگر در ناحیه‌ی بحرانی خودش به هیچ وجه نخواهد رفت و سپس برای ورود به ناحیه‌ی بحرانی خودش اقدام کند. ایده‌ی روش سوم از همین زاویه دید نشأت گرفته است. یعنی: یک فرآیند، در وهله اول به فرآیند های دیگر اجازه‌ی ورود نمی‌دهد و سپس اجازه‌ی ورود خودش را بررسی می‌کند. این تغییر به سادگی و فقط با جابه‌جا کردن دو خط از روش دوم میسر می‌شود. چرا دیگر مشکل قبضه کردن را نخواهیم داشت؟ چون ابتدا فرآیند دیگر منع می‌شود از ورود به ناحیه‌ی بحرانی.

(ب) تفاوت گرسنگی با بن بست را توضیح دهید.

بن بست (Deadlock): شرایطی است که چند فرآیند در رقابت برای دسترسی به منابع موردنیازشان تا ابد اجازه‌ی پیش‌روی نداشته باشند. در این شرایط تعامل بین چندفرآیند (نه فقط یکی) است که بن بست را رقم زده است. در ضمن این عدم دسترسی به منبع موردنیاز تا ابد ادامه پیدا می‌کند اگر تمهیدی برای آن در نظر گرفته نشده باشد. **گرسنگی (Starvation):** وقتی یک فرآیند تحت شرایطی (احتمالاتی/قطعی) امکان دسترسی به یک منبع را تا زمانی مشخص به دست نیآورد.

تفاوت کلیدی بین مفهوم گرسنگی و بن بست در اینست که گرسنگی حالتی محدودتر و با پیود بیشتر از گرسنگی است. در بن بست چندفرآیند تا ابد به مشکل دسترسی به منابع برمیخورند. وقوع یک بن بست به معنی وقوع یک گرسنگی است. چون حداقل یک فرآیند داریم که تا ابد به منبع موردنیازش دسترسی پیدا نخواهد کرد. برعکس این گزاره لزوماً برقرار نیست. یعنی ممکن است گرسنگی رخ بدهد و بن بست رخ نداده باشد.

(ج) دستورالعمل TSL و دستورالعمل Swap را مقایسه کنید.

دستور TSL: یک عملیات اتمیک (تقسیم ناپذیر) است. به این معنی که یا هر دو دستور مذکور اجرا میشوند یا هیچکدام اجرا نمیشوند (یا همه یا هیچ). ابتدا محتویات یک خانه از حافظه را خوانده و در یک ثبات ذخیره میکند. سپس مقدار یک را در دقیقاً همان آدرس از حافظه ذخیره میکند.

```
tsl reg, lock
```

دستور swap: این دستور فقط از یک عمل تقسیم ناپذیر تشکیل شده است. محتویات یک رجیستر را با محتویات یک خانه از حافظه‌ی اصلی جابه‌جا می‌کند.

```
swap reg, lock
```

نکته: دستورات ذکر شده خود دارای یک سری عملیات‌های کوچک‌تر به صورت ضمنی هستند و اجرای این مجموعه دستورات ضمنی است که اتمیک است.

(د) زمان بندی SRT چه ویژگی هایی دارد و تفاوت آن با SJF چیست؟

این دو زمانبندی بسیار به هم شباهت دارند. ایده‌ی اصلی در قبضه‌ای یا غیرقبضه‌ای بودن آنها است. در SJF زمان‌بند، فرآیند با کوتاه‌تری زمان انفجار را انتخاب می‌کند و سپس پردازنده را به آن اختصاص می‌دهد. این زمان بندی از نوع غیر قبضه‌ای است. یعنی فرآیند تا زمان خاتمه یافتن به اجرای خودش ادامه می‌دهد. روش پیاده سازی این روش با درخت هرم کمینه (min-heap tree) به سادگی میسر می‌شود.

از طرفی SRT همان روش قبلی با پیاده‌سازی پیچیده‌تر و عملکردی منعطف تر است. این مقایسه‌ی زمان انفجار پیوسته در حال انجام است. دو نکته را باید در پیاده سازی مد نظر قرار داد و آن را مدیریت کرد:

- فرآیند در حال اجرا که به صورت پیوسته از زمان انفجارش کم می‌شود.
- یک فرآیند جدید وارد صف آماده شود.

برخلاف روش قبل که پردازنده از فرآیند تا اتمام آن یا وقوع وقفه حین اجرا قبضه نمی‌شد، هنگام ورود به یک فرآیند جدید، درخت هرم کمینه بروزرسانی می‌شود و در پایان این بروزرسانی فرآیند با کمترین زمان انفجار در ریشه‌ی درخت است و زمان بند آن را انتخاب کند و پردازنده را به آن اختصاص دهد. این فرآیند ممکن است که فرآیند در حال اجرا و یا فرآیند تازه وارد باشد.

تمرین ۳

سیستمی را که شامل چهار فرآیند همزمان و دو منبع قابل استفاده مجدد است در نظر بگیرید. به شرط اینکه هر فرآیند حداکثر به دو منبع نیاز داشته باشد، تعداد وضعیت های بن بست (Deadlock States) در این سیستم حداکثر چند حالت است؟

بن بست در یک گروه از فرآیندهای همکار بدین معنا است که هیچ کدام از آنها از منابع اشتراکی نمی‌توانند استفاده کنند چرا که در هر فرآیند منتظر فرآیند های دیگر می‌ماند. البته برای این که بن بست رخ دهد این انتظار باید نامحدود باشد. با توجه به فرض بالا در صورتی که یک فرآیند به هیچ منبع اشتراکی احتیاج نداشته باشد، مستقل از سایر فرآیند هاست و در اینجا مورد بررسی قرار نمی‌گیرد ولی اگر آن را در نظر بگیریم در واقع بن بست نمی‌خواهیم داشت. حال اگر فرض کنیم تعدادی از این فرآیندها به یک منبع و تعدادی دیگر به دو منبع اشتراکی برای اجرا نیاز داشته باشند، شرایط بن بست را مورد بررسی قرار می‌دهیم.

فرض می‌کنیم در یک لحظه دو فرآیند که هر کدام به هر دو منبع اشتراکی نیاز دارند، تنها یکی از آنها را در اختیار داشته باشند و تا زمانی که منبع دیگر را در اختیار نگیرند این منبع را رها نکنند (در حالت های دیگر بن بست رخ نخواهد داد و در بدترین حالت livelock خواهیم داشت). تعداد حالت های رخ دادن بن بست با فرض این که تعداد فرآیندهایی که به هر دو احتیاج دارند x و حداکثر چهار باشد، $\binom{x}{2}$ خواهد بود.

تمرین ۴

سه فرآیند P_1, P_2, P_3 با زمان اجرا و زمان ورود زیر را در نظر بگیرید و به موارد خواسته شده پاسخ دهید. (عدد بالاتر بیانگر اولویت بالاتر است.)

فرآیند	اولویت	زمان ورود	زمان اجرا
P_1	۲	t	۴
P_2	۰	t	۲
P_3	۱	$t + 3$	۱

(آ) متوسط زمان پاسخگویی با روش SJF

$$\begin{array}{ll}
 wT(P_1) = t + 2 - t = 2 & T(P_1) = 2 + 4 = 6 \\
 wT(P_2) = t - t = 0 & T(P_2) = 0 + 2 = 2 \\
 wT(P_3) = t + 6 - (t + 3) = 3 & T(P_3) = 3 + 1 = 4
 \end{array}$$

P_2	P_1	P_3
t	$t + 2$	$t + 6$
$t + 7$		

$$AvgT = \frac{6 + 2 + 4}{3} = 4$$

(ب) متوسط زمان پاسخگویی با روش FIFO

$$\begin{array}{ll}
 wT(P_1) = t - t = 0 & T(P_1) = 0 + 4 = 4 \\
 wT(P_2) = t + 4 - t = 4 & T(P_2) = 4 + 2 = 6 \\
 wT(P_3) = t + 6 - (t + 3) = 3 & T(P_3) = 3 + 1 = 4
 \end{array}$$

P_2	P_1	P_3
t	$t + 4$	$t + 6$
$t + 7$		

$$AvgT = \frac{4 + 6 + 4}{3} = \frac{14}{3}$$

(ج) متوسط زمان پاسخگویی با روش SRT

$$\begin{array}{ll}
 wT(P_1) = (t + 2) - t + \\
 t + 4 - (t + 3) = 3 & T(P_1) = 3 + 4 = 7 \\
 wT(P_2) = t - t = 0 & T(P_2) = 0 + 2 = 2 \\
 wT(P_3) = t - t = 0 & T(P_3) = 0 + 1 = 1
 \end{array}$$

P_2	P_1	P_3	P_1
t	$t + 2$	$t + 3$	$t + 4$
$t + 7$			

$$AvgT = \frac{7 + 2 + 1}{3} = \frac{10}{3}$$

(د) متوسط زمان پاسخگویی با روش اولویت

$$\begin{array}{ll}
 wT(P_1) = t - t = 0 & T(P_1) = 0 + 4 = 4 \\
 wT(P_2) = t + 5 - t = 5 & T(P_2) = 5 + 2 = 7 \\
 wT(P_3) = t + 4 - (t + 3) = 1 & T(P_3) = 1 + 1 = 2
 \end{array}$$

P_1	P_3	P_2
t	$t + 4$	$t + 5$
$t + 7$		

$$AvgT = \frac{4 + 7 + 2}{3} = \frac{13}{3}$$

تمرین ۵

یک سیستم تک پردازنده از روش زمان بندی صف چند سطحی (MFQ) استفاده می کند که در آن صف سطح اول و دوم از زمان بندی چرخشی (RR) به ترتیب با کوانتوم زمانی ۴ و ۸ میلی ثانیه با زمان تعویض متن ۱ میلی ثانیه و سطح سوم از روش FCFS استفاده می نماید. چهار فرآیند با زمان اجرای ۱۰ و ۳ و ۷ و ۲ به ترتیب در زمان های ۰ و ۳ و ۷ و ۲۴ وارد می شوند. میانگین زمان پاسخ و زمان انتظار برای اجرای کامل فرآیندها را محاسبه کنید. ابتدا صف ها را نامگذاری می کنیم:

صف با زمانبندی RR کوانتوم تایم ۴ میلی ثانیه $A \leftarrow$

صف با زمانبندی RR کوانتوم تایم ۸ میلی ثانیه $B \leftarrow$

صف با زمانبندی FCFS (صف آخر) $C \leftarrow$

فرآیند	زمان ورود	زمان انفجار
P_0	۰	۱۰
P_1	۳	۳
P_2	۷	۷
P_3	۲۴	۲

P_0		P_1		P_2		P_0		P_2		P_3	
۰	۴	۵	۸	۹	۱۳	۱۴	۲۰	۲۱	۲۴	۲۵	۲۷

- ابتدا در زمان صفر فقط فرآیند P_0 در صف A وجود دارد. پس توسط زمان بند این صف انتخاب می شود.
- حین اجرای P_0 فرآیند P_1 وارد صف A می شود اما آن را قبضه نمی کند.
- در زمان ۴ اجرای P_0 تمام می شود. این فرآیند به صف B منتقل می شود.
- در صف A فرآیند P_1 باقی مانده پس توسط زمان بند این صف انتخاب می شود. این فرآیند در همین مرحله خاتمه می یابد و از صف های آماده خارج می شود.
- اولویت صف A بیشتر است پس P_2 توسط زمان بند این صف انتخاب شده و ۴ میلی ثانیه اجرا می شود
- در زمان ۸، فرآیند P_0 در صف B است اما در زمان ۷ فرآیند P_2 وارد صف A شده است.
- سپس فرآیند P_2 به صف B منتقل می شود. الان در این صف دو فرآیند P_0 و P_2 وجود دارند. چون زمان بندی از نوع RR است، پس فرآیندی که بیشتر در صف B منتظر مانده است انتخاب می شود. یعنی P_0 انتخاب می شود و در کوانتوم این صف اجرا می شود.
- سپس این فرآیند از این صف خارج می شود. فقط فرآیند P_2 را در صف B داریم. این فرآیند توسط زمان بند این صف انتخاب می شود و اجرا می شود.
- زمان ۲۴ فرآیند جدید P_3 وارد صف آماده می شود. این فرآیند وار صف A می شود. توسط زمان بند این صف انتخاب و اجرا می شود.

تمرین ۶

فرض کنید در کد زیر p1 و p2 همروند باشند و سمافورهای x و y دو سمافور دودویی و هر دو مقدار اولیه یک را داشته باشند. همچنین فرض کنید دستور print در یک چرخه دستورالعمل اجرا شود.

p2:	p1:
----	----
wait(y)	wait(x)
while true do	while true do
print "B"	print "A"
signal(y)	signal(x)
----	----

آیا هر تلفیقی از A ها و B ها ممکن است در خروجی ظاهر شوند؟ توضیح دهید.
با توجه به اینکه این دو فرآیند از سمافورهای متفاوتی استفاده می‌کنند هیچ کدام منتظر دیگری نمی‌ماند. با فرض اینکه این دو فرآیند همروند اجرا شوند، بنابراین در هر قسمتی از اجرا ممکن است اجرای آن‌ها متوقف شود و CPU به فرآیند دیگر اختصاص داده شود که این بدان معناست که خروجی این برنامه‌ها از الگوی خاصی پیروی نمی‌کند و هر حالتی ممکن است داشته باشد.