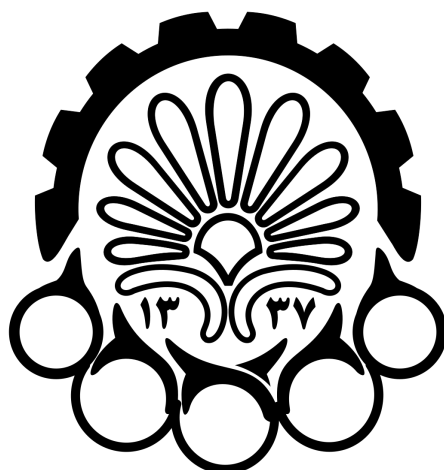


سیستم‌های عامل
دکتر جوادی



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

رضا آدینه پور ۴۰۲۱۳۱۰۵۵

تمرین سری چهارم

۷ دی ۱۴۰۲



سوال اول

چهار پردازش زیر به صورت موازی شروع به اجرا می‌کنند، اگر مقدار اولیه هر سمافور غیر باینری مطابق با جدول زیر باشد، با ذکر دلیل نشان دهید کدام رشته‌ها امکان چاپ ندارند و برای رشته‌هایی که امکان چاپ دارند ترتیب اجرای دستورات پردازش را ذکر کنید.

P1:	P2:	P3:	P4:
<pre>while (true) { wait(s1); print("A"); signal(s2); print("B"); }</pre>	<pre>while (true) { wait(s2); print("B"); signal(s3); print("C"); }</pre>	<pre>while (true) { signal(s3); print("C"); wait(s1); print("A"); }</pre>	<pre>while (true) { wait(s3); print("A"); signal(s1); print("B"); wait(s2); print("C"); }</pre>

initial value:
s1 = 0
s2 = 0
s3 = 0

۱. AACBB
۲. BABAB
۳. CABAA
۴. CBAAC

پاسخ

در ابتدا بررسی می‌کنیم که کدام رشته‌ها قادر به چاپ هستند:

۱. P_1 : مقدار اولیه s_2 صفر است، بنابراین در خط اول $wait(s_2)$ پردازش مسدود می‌شود و هرگز به خط چاپ "A" نمی‌رسد. پس امکان چاپ ندارد.

۲. P_2 : مقدار اولیه s_2 صفر است، بنابراین در خط اول $wait(s_2)$ پردازش مسدود می‌شود و هرگز به خط چاپ "B" نمی‌رسد. پس امکان چاپ ندارد.

۳. P_3 : مقدار اولیه s_1 صفر است، بنابراین در خط سوم $wait(s_1)$ پردازش مسدود می‌شود و هرگز به خط چاپ "A" نمی‌رسد. پس امکان چاپ ندارد.

۴. P_4 : هیچ $wait$ اولیه‌ای ندارد، بنابراین پردازش می‌تواند به اجرای خطوط بعدی بپردازد. پس امکان چاپ دارد.

بنابراین، تنها رشته‌ای که امکان چاپ دارد P_4 است.

۱. P_4 : بدون هیچ مسدودی، خطوط کد به ترتیب اجرا می‌شوند. بنابراین چاپ "A"، سیگنال دادن به s_1 ، چاپ "B"، انتظار برای s_2 ، چاپ "C" رخ می‌دهد. پس ترتیب اجرای دستورات: ABC است

سوال دوم

چه تعداد خروجی ۱۰ کاراکتره برای دو پردازش زیر که موازی با یکدیگر اجرا می‌شود ممکن است؟ کلیت الگو آن را ذکر کنید.
(مقدار اولیه هر دو سمافور ۰ و غیر باینری هستند)

<p>P1:</p> <pre>while (true) { signal(S1); print("A"); wait(S2); print("A"); }</pre>	<p>P2:</p> <pre>while (true) { signal(S2); print("B"); wait(S1); print("B"); }</pre>
--	--

راهنمایی: از اصل جمع و حالت بندی بر اساس وضعیت ۴ کاراکترهای چاپ شده استفاده کنید.

پاسخ

برای تعیین تعداد خروجی‌های ۱۰ کاراکتره برای دو پردازش P1 و P2 با استفاده از اصل جمع و حالت بندی بر اساس وضعیت چهار کاراکتر چاپ شده، به طور مشابه با پرسش قبل عمل می‌کنیم. در اینجا نیز، دو کاراکتر "A" و "B" را در نظر می‌گیریم.
حالت‌های ممکن برای چهار کاراکتر چاپ شده عبارتند از:

۱. AABB: P1 یک بار چاپ "A" و سپس P2 یک بار چاپ "B"

۲. ABAB: P1 یک بار چاپ "A" و سپس P2 یک بار چاپ "B"

۳. BAAB: P2 یک بار چاپ "B" و سپس P1 یک بار چاپ "A"

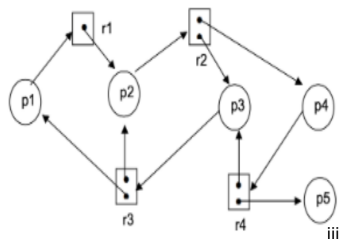
۴. BBAA: P2 یک بار چاپ "B" و سپس P1 یک بار چاپ "A"

با توجه به اصل جمع و حالت بندی، تعداد کل خروجی‌های ۱۰ کاراکتره برابر است با تعداد حالت‌های ممکن برای چهار کاراکتر چاپ شده، که در اینجا برابر با ۴ است.

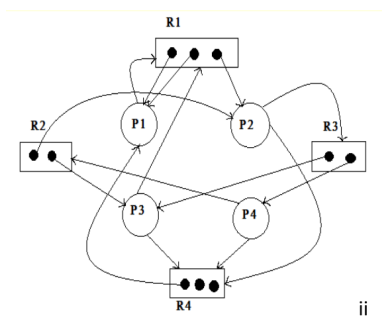
بنابراین، تعداد خروجی‌های ۱۰ کاراکتره برای دو پردازش P1 و P2 برابر با ۴ است و الگوی کلی آن عبارت است از: BBAA, BAAB, ABAB, AABB.

سوال سوم

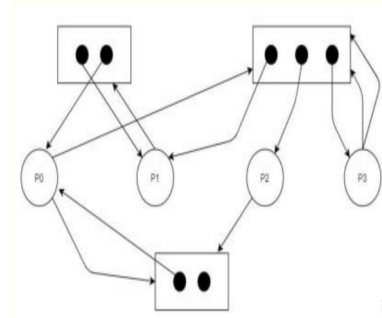
گراف‌های تخصیص منابع زیر را در نظر بگیرید، مشخص کنید آیا سیستم بن‌بست دارد یا خیر، اگر دارد دلیل خود را بیان کنید و اگر نه یک دنباله از اجرای فرایندها بنویسید.



(ج) گراف ۳



(ب) گراف ۲



(آ) گراف ۱

شکل ۱: گراف‌ها

سوال چهارم

سیستم زیر را در نظر بگیرید:

All Recourse A B C D	Max A B C D	Allocation B C D	
3 14 12 12	0 0 1 2	0 0 1 2	P0
	1 7 5 0	1 0 0 0	P1
	2 3 5 6	1 3 5 4	P2
	0 6 5 2	0 6 3 2	P3
	0 6 5 6	0 0 1 4	P4

۱. ماتریس need را تشکیل دهید.

۲. آیا سیستم در حالت safe قرار دارد؟

۳. اگر پردازنده P1 درخواست ۰،۲،۴،۰ را بدهد، آیا می‌توان بلافاصله به آن پاسخ داد؟

پاسخ

۱. ماتریس need به صورت زیر است:

$$\text{Need} = \begin{bmatrix} 3 & 14 & 11 & 10 \\ 1 & 7 & 5 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 6 & 4 & 2 \end{bmatrix}$$

۲. بله سیستم در حالت safe قرار دارد.

$$\text{Available} = (0, 0, 1, 2)$$

۳. با توجه به اینکه نیازها از منابع موجود کمتر هستند، درخواست بلافاصله پاسخ داده می‌شود.

$$\text{Available} = (0, 0, 1, 2)$$

$$\text{Allocation}_{P1} = (1, 0, 0, 0)$$

$$\text{Need}_{P1} = (0, 7, 5, 0)$$

سوال پنجم

۱. با توجه به ۵ پارتیشن حافظه ۱۰۰ کیلوبایتی، ۵۰۰ کیلوبایتی، ۲۰۰ کیلوبایتی، ۳۰۰ کیلوبایتی و ۶۰۰ کیلوبایتی (به ترتیب) هریک از الگوریتم‌های bestFit و firstFit و worstFit به ترتیب اندازه‌های ۱۱۲، ۴۱۷، ۲۱۲ و ۴۲۶ کیلوبایتی را چگونه قرار می‌دهند و بگویید کدام یک از الگوریتم‌های ذکر شده بهینه‌تر است؟

۲. با توجه به جدول زیر، آدرس فیزیکی مربوط به آدرس‌های منطقی زیر را به دست آورید.

- (a) 0, 430
- (b) 1, 10
- (c) 2, 500
- (d) 3, 400
- (e) 4, 112

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1954	96

پاسخ

۱. برای قرار دادن اندازه‌ها با اندازه‌های ۱۱۲، ۴۱۷، ۲۱۲ و ۴۲۶ کیلوبایت در الگوریتم‌های bestFit و firstFit و worstFit، می‌توان به صورت زیر عمل کرد:

(آ) اندازه ۱۱۲ کیلوبایتی:

- i. bestFit: بلوک ۱ با بازه بین ۲۰۰ تا ۲۹۹ کیلوبایت
- ii. firstFit: بلوک ۱ با بازه بین ۲۰۰ تا ۲۹۹ کیلوبایت
- iii. worstFit: بلوک ۳ با بازه بین ۴۰۰ تا ۵۹۹ کیلوبایت

(ب) اندازه ۴۱۷ کیلوبایتی:

- i. bestFit: بلوک ۴ با بازه بین ۱۱۲ تا ۲۲۷ کیلوبایت
- ii. firstFit: بلوک ۱ با بازه بین ۲۰۰ تا ۵۹۹ کیلوبایت
- iii. worstFit: بلوک ۳ با بازه بین ۴۰۰ تا ۵۹۹ کیلوبایت

(ج) اندازه ۲۱۲ کیلوبایتی:

- i. bestFit: بلوک ۱ با بازه بین ۲۰۰ تا ۲۹۹ کیلوبایت
- ii. firstFit: بلوک ۱ با بازه بین ۲۰۰ تا ۲۹۹ کیلوبایت
- iii. worstFit: بلوک ۳ با بازه بین ۴۰۰ تا ۵۹۹ کیلوبایت

(د) اندازه ۴۲۶ کیلوبایتی:

- i. bestFit: بلوک ۵ با بازه بین ۹۶ تا ۱۹۱ کیلوبایت
- ii. firstFit: بلوک ۱ با بازه بین ۲۰۰ تا ۵۹۹ کیلوبایت
- iii. worstFit: بلوک ۳ با بازه بین ۴۰۰ تا ۵۹۹ کیلوبایت

پاسخ

۱. برای به دست آوردن آدرس فیزیکی مربوط به آدرس‌های منطقی زیر، می‌توانیم از جدول Base Segment و Length استفاده کنیم:

$$\text{آدرس منطقی } (0, 430): \text{ آدرس فیزیکی } = \text{Offset} + \text{Base} \\ \text{آدرس فیزیکی } = 219 + 430 = 649$$

$$\text{(ب) آدرس منطقی } (1, 10): \text{ آدرس فیزیکی } = 2300 + 10 = 2310$$

$$\text{(ج) آدرس منطقی } (2, 500): \text{ آدرس فیزیکی } = 500 + 90 = 590$$

$$\text{(د) آدرس منطقی } (3, 400): \text{ آدرس فیزیکی } = 1327 + 400 = 1727$$

$$\text{(ه) آدرس منطقی } (4, 112): \text{ آدرس فیزیکی } = 1954 + 112 = 2066$$

سوال ششم

در این تمرین می‌خواهیم با مدیریت حافظه اصلی در لینوکس آشنا شویم. در لینوکس همه اطلاعات به صورت فایل ذخیره شده است. به طور مثال درون دایرکتوری `/proc` اطلاعات مربوط به هر پردازش و منابع سیستم موجود است. در این دایرکتوری، به ازای هر پردازش موجود در سیستم یک دایرکتوری با شماره PID آن پردازش ساخته شده است. که اطلاعات مربوط به آن پردازش را نگهداری می‌کند. همچنین اطلاعاتی مانند اطلاعات مربوط به حافظه اصلی و... را می‌توان در این دایرکتوری یافت. محتوای این فایل را نمایش دهید. برای اینکه ببینیم چقدر از حافظه اصلی استفاده شده است، از دستور `free` نیز می‌توان استفاده کرد.

پاسخ

به دایرکتوری گفته شده می‌رویم و با دستور `cat meminfo` محتویات فایل را می‌خوانیم

```

reza@r324:/proc
+ /proc ls
1 1126 1515 1698 190 2059 23796 25702 27219 27720 304 3229 3681 3763 4002 49 59 74 86 asound ionem mtrr version
101 113 1516 1699 191 21 23818 26 27264 27743 305 33 3691 3791 4010 5 6 741 862 bootconfig net net version signature
102 114 1517 17 192 2140 24950 26424 27278 27772 306 3329 37 38 4022 50 61 75 87 buddyinfo irq pagetypeinfo vmallocinfo
1025 115 16 1706 193 2190 24956 26506 27279 27776 3072 3340 3700 3801 4054 5094 62 76 89 bus kallsyms partitions vmstat zoneinfo
103 116 164 1718 1942 22 25 26507 27328 27779 3099 34 3715 3843 4074 51 6246 7682 90 cgroups kcore pressure
104 118 1658 1724 1943 220 250 26607 27332 27783 31 3468 3716 3844 4076 52 63 77 91 cmdline keys schedstat
105 119 1653 1725 1944 221 25027 26610 27350 27809 3103 3490 3720 3845 4078 5296 64 771 93 consoles key-users scsi
106 12 1655 1726 1951 222 25147 26695 27387 27843 3110 35 3721 3851 41 53 65 772 94 cpuinfo knsp self
107 120 1656 1727 1962 223 25255 26711 27438 27850 3111 3505 3725 3862 4115 5319 67 7740 95 kpagecgroup slabinfo
1071 127 1657 1748 1964 2252 25258 26712 27443 27927 3115 3532 3734 3869 4220 5357 68 7761 952 devices kpagecount softirqs
1073 13 1658 1750 1976 2253 25272 268 27515 27941 3120 3533 3736 3871 4221 5434 69 79 96 diskstats kpageflags stat
1074 1342 1668 18 1977 2275 25273 26858 27516 27946 3130 3538 3741 39 4238 55 70 7923 97 dma loadavg swaps
1075 1343 167 1833 2 23 25274 26866 27536 27948 3132 3545 3744 3902 43 56 7031 8 9794 driver locks sys
1076 1350 1672 1854 20 23286 25276 26892 27613 27969 3134 3549 3746 3923 4376 5612 71 80 98 dynamic_debug mdstat sysrq-trigger
1077 1381 1675 186 2006 2332 25277 27 27622 27981 3168 3558 3750 3954 44 5630 728 81 9806 execdomains meminfo sysvipc
109 1382 1689 187 2014 2333 25278 27001 27624 28 3178 3561 3751 3980 45 57 73 82 9809 fb thread-self
11 14 169 188 20154 2336 25302 27028 27634 29 3186 3589 3753 3995 46 5740 733 823 9812 filesystems misc
110 141 1691 189 2019 2338 25328 27057 27693 3 32 3620 3760 4 47 58 734 83 99 fs modules timer_list
111 15 1697 19 20348 2341 254 27204 27698 301 3226 3653 3762 40 4874 5889 735 85 acpi mounts tty
+ /proc free
total used free shared buff/cache available
Mem: 12084800 5160536 3077340 812648 4980492 6924264
Swap: 4194300 0 4194300
+ /proc

```

```

reza@r324:/proc
+ /proc cat meminfo
MemTotal: 12084796 kB
MemFree: 4508412 kB
MemAvailable: 7591228 kB
Buffers: 204008 kB
Cached: 4847644 kB
SwapCached: 0 kB
Active: 4448292 kB
Inactive: 1448212 kB
Active(anon): 2642448 kB
Inactive(anon): 11940 kB
Active(file): 1805844 kB
Inactive(file): 1436272 kB
Unevictable: 961260 kB
Mlocked: 48 kB
SwapTotal: 4194300 kB
SwapFree: 4194300 kB
Zswap: 0 kB
Zswapped: 0 kB
Dirty: 724 kB
Writeback: 0 kB
AnonPages: 2603664 kB
Mapped: 1011424 kB
Shmem: 1009536 kB
KReclaimable: 162956 kB
Slab: 430572 kB
SReclaimable: 162956 kB
SUnreclaim: 267616 kB
KernelStack: 21088 kB
PageTables: 39004 kB
SecPageTables: 0 kB
NFS_Unstable: 0 kB
Bounce: 0 kB
WritebackTmp: 0 kB
CommitLimit: 10236696 kB
Committed_AS: 12392668 kB
VmallocTotal: 34359738367 kB
VmallocUsed: 126796 kB
VmallocChunk: 0 kB
Percpu: 16576 kB
HardwareCorrupted: 0 kB
AnonHugePages: 8192 kB
ShmemHugePages: 0 kB
ShmemPmdMapped: 0 kB
FileHugePages: 0 kB
FilePmdMapped: 0 kB
Unaccepted: 0 kB
HugePages_Total: 0

```

در خروجی فایل `meminfo`، ستون‌های مختلف به معنای زیر هستند:

پاسخ

۱. MemTotal: مجموع حافظه اصلی سیستم به کیلوبایت.
 ۲. MemFree: مقدار حافظه اصلی خالی و در دسترس به کیلوبایت.
 ۳. MemAvailable: مقدار حافظه اصلی در دسترس برای پراسس‌ها به کیلوبایت.
 ۴. Buffers: حافظه‌ای که برای نگهداری داده‌های برنامه‌های I/O استفاده می‌شود.
 ۵. Cached: حافظه‌ای که برای نگهداری داده‌هایی که توسط سیستم فایل‌های کش شده‌اند، استفاده می‌شود.
 ۶. SwapCached: حافظه‌ای که در بخش فایل‌های صفحه‌بندی (swap) برای فایل‌های کش شده استفاده شده است.
 ۷. Active: حافظه‌ای که در حال استفاده توسط فرآیندهای فعال است.
 ۸. Inactive: حافظه‌ای که در حال حاضر توسط فرآیندهای غیرفعال استفاده نمی‌شود.
 ۹. ...
- دستور `free -h` را اجرا کرده و توضیح دهید که هر یک از ستون‌های آن به چه معناست؟
خروجی دستور `free -h`:

```
reza@r3z4:/proc
→ /proc free -h
total      used      free      shared    buff/cache   available
Mem:       11Gi      5.1Gi      2.8Gi      836Mi      4.8Gi        6.5Gi
Swap:      4.0Gi          0B      4.0Gi
```

با توجه به خروجی بالا، ستون‌های مختلف در دستور `free -h` به معنای زیر هستند:

۱. total: مجموع حافظه فیزیکی (RAM) و حافظه اصلی سیستم.
۲. used: مقداری از حافظه فیزیکی که در حال استفاده است.
۳. free: مقداری از حافظه فیزیکی که در حال حاضر خالی و در دسترس است.
۴. shared: حافظه فیزیکی مشترک بین پردازنده‌ها.
۵. buff/cache: مقداری از حافظه فیزیکی که به عنوان بافر و کش استفاده می‌شود.
۶. available: حافظه فیزیکی در دسترس برای پراسس‌ها.

پاسخ

برای اینکه تغییرات حافظه اصلی را دنبال کنیم، می‌توان از دستور `watch free -h` استفاده کرد.

```

watch free -h
Every 2.0s: free -h                               r3z4: Mon Dec 25 20:41:56 2023

```

	total	used	free	shared	buff/cache	ava
Mem:	11Gi	5.3Gi	2.5Gi	817Mi	4.8Gi	
Swap:	4.0Gi	0B	4.0Gi			

همانطور که در خروجی دستور `free` می‌بینید، حجم زیادی از حافظه اصلی به بافر و `cache` اختصاص داده شده است. برای پاک کردن فضای اشغال شده توسط `cache` می‌توان از دستور زیر استفاده کرد.

```
sudo sh -c "sync; echo 3 > /proc/sys/vm/drop_caches"
```

دستور فوق را تفسیر کنید و دوباره با دستور `free` مقدار ستون `buff/cache` را نشان دهید:

```

reza@r3z4:/proc
→ /proc sudo sh -c "sync; echo 3 > /proc/sys/vm/drop_caches"
→ /proc free -h

```

	total	used	free	shared	buff/cache
Mem:	11Gi	5.3Gi	5.4Gi	770Mi	1.8Gi
Swap:	4.0Gi	0B	4.0Gi		

پس از اجرای دستور فوق، مقدار ستون `buff/cache` به 1.8 G کاهش پیدا کرد. تفسیر این دستور به صورت زیر است:

۱. `sudo`: استفاده از دستورات با دسترسی ادمین (`superuser`).

۲. `sh -c`: اجرای یک دستور شل

۳. `"sync; echo 3 > /proc/sys/vm/drop_caches"`

در این بخش، دو دستور اجرا می‌شود. ابتدا دستور `sync` اجرا می‌شود که اطلاعات موجود در حافظه نهان (`cache`) را بر روی دیسک ذخیره می‌کند. سپس دستور

```
echo 3 > /proc/sys/vm/drop_caches
```

اجرا می‌شود که باعث پاک شدن محتوای حافظه نهان (`cache`) می‌شود. عدد ۳ به معنای پاک کردن بخش‌های `cache`، مشترک و `inode` است.