پاییز ۱۳۹۴

تمرین سری پنجم

سیستم‌های عامل

پرهام الوانی

۹۲۳۱۰۵۸

سوال ۱

در SJF از آنجایی که همواره پردازه‌ای اجرا میشود که کمترین نیاز به پردازنده را دارد، زمان انتظار برای سایر پردازه‌ها به حداقل می‌رسد.

سوال ۲

FCFS

	P1	P2	P3	P4	P5	T
Turn Around	10	20	14	28	7	15.8
Waiting	0	5	13	14	0	6.4

RR

	P1	P2	P3	P4	P5	T
Turn Around	17	30	4	28	7	17.2
Waiting	7	15	3	13	0	7.6

SJF

	P1	P2	P3	P4	P5	T
Turn Around	10	21	1	28	7	13.4
Waiting	0	6	0	13	0	3.8

	P1	P2	P3	P4	P5	T
Turn Around	20	45	4	23	7	20.4
Waiting	10	30	3	7	0	10.6

سوال ۳

الگوریتم JSF در صورتی که پروسس‌های ورودی همگی نیاز کمی به پردازنده داشته باشند، آنها را اجرا کرده و پردازه‌ای که نیاز بیشتری به پردازنده دارد دچار قحطی میشود.

سوال ۴

الگوریتم CFS یا Completely Fair Scheduler در کرنل لینوکس یک الگوریتم زمان بند هست که به پروسس‌ها الویت میدهد و از درخت قرمز-سیاه استفاده کرده و در زمان O(1) عمل میکند.

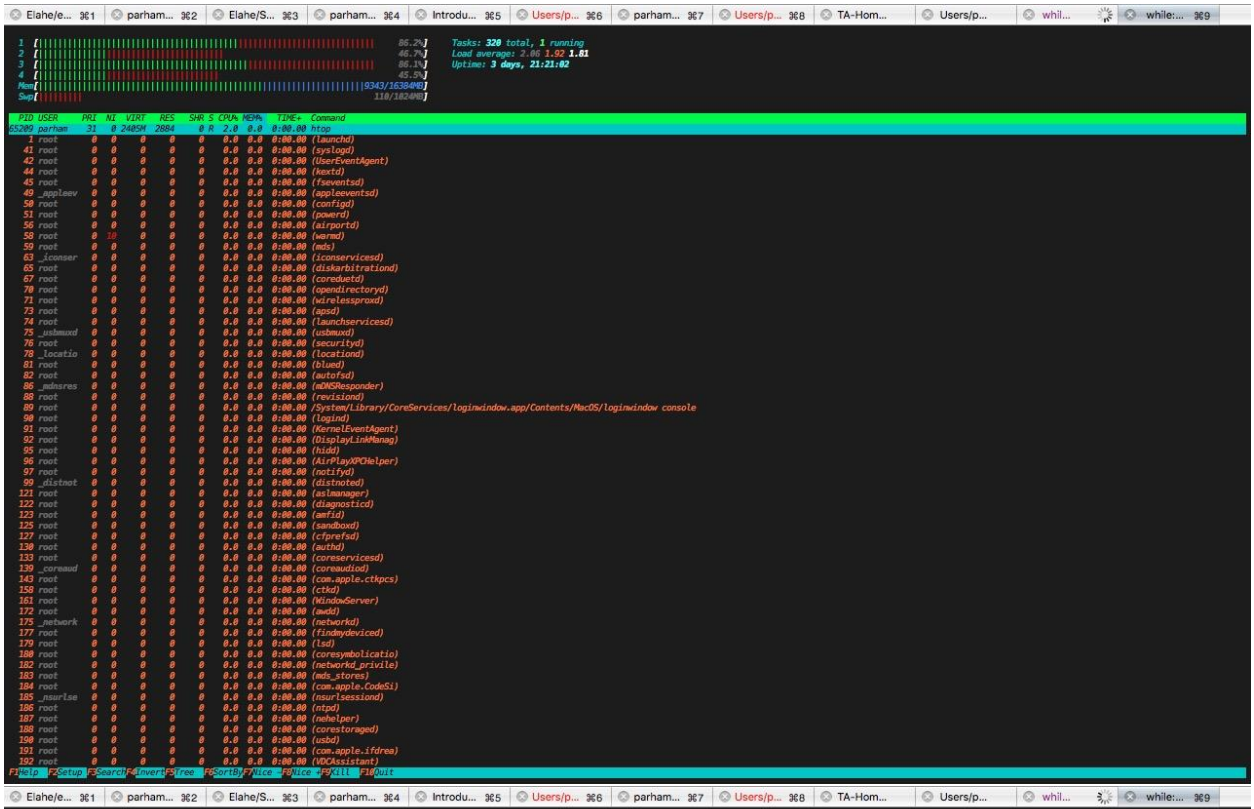
سوال ۵

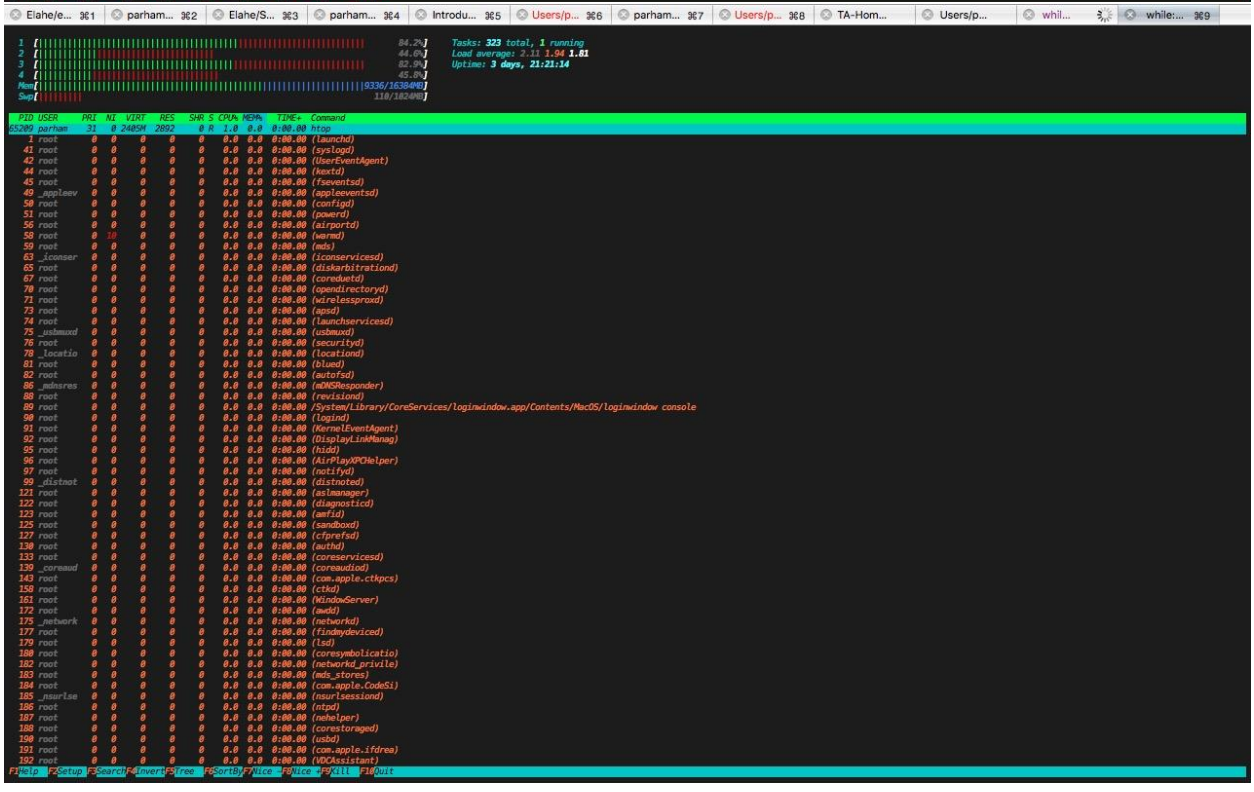
زمانبد باید به پروسس‌هایی که استفاده زیادی از CPU دارند زمان بیشتری را اختصاص دهد این در حالی است که باید به پروسس‌هایی که استفاده کمتری از CPU داشته و بیشتر در انتظار IO هستند زمان کمتری از CPU را اختصاص دهد. اگر زمانبند نتواند این کار را به درستی انجام دهد مقدار زیادی از زمان CPU تلف خواهد شد.

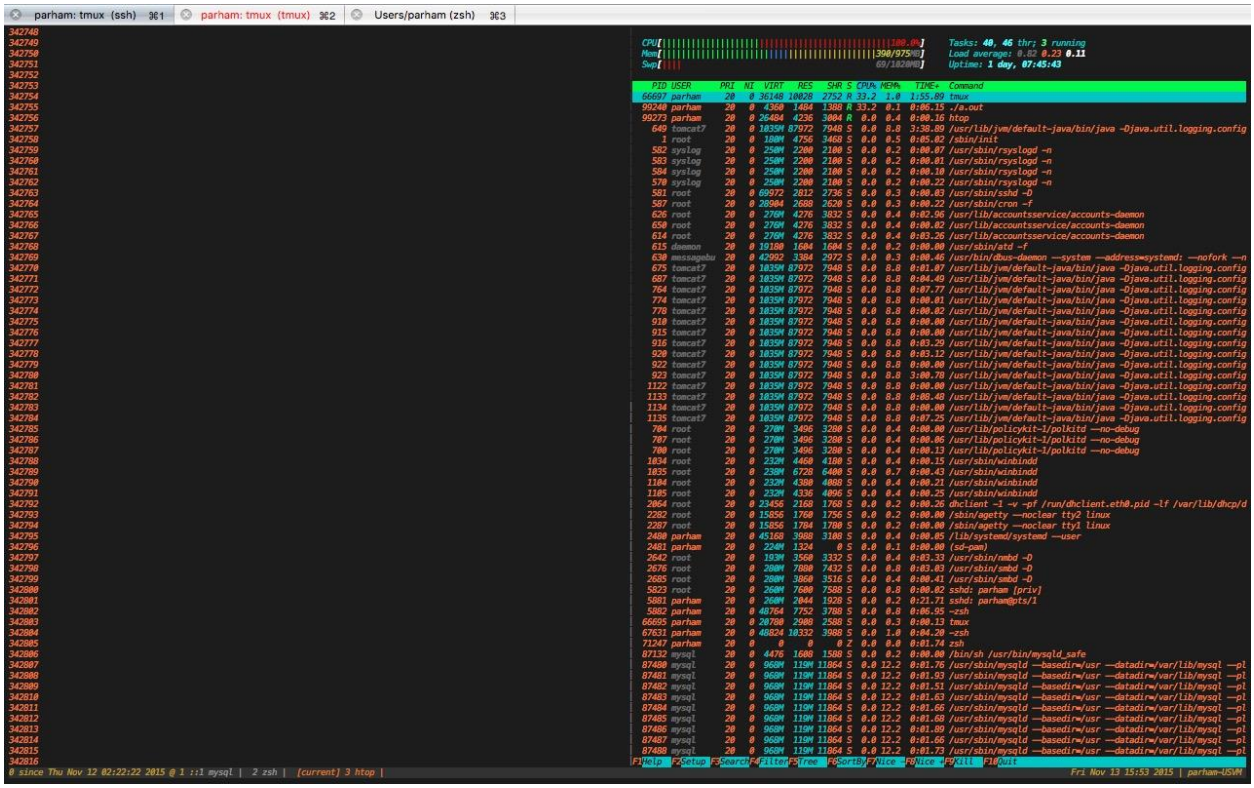
سوال ۶

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
    int i = 0;
    while (1) {
        i++;
        printf("%d\n", i);
    }
    return 0;
}
```

```c
#define _GNU_SOURCE

#include <stdio.h>
#include <sched.h>

int main(int argc, char *argv[])
{
    int i, ret;
    cpu_set_t set;

    CPU_SET(0, &set);
    ret = sched_setaffinity(0, sizeof(cpu_set_t), &set);
    if (ret == -1)
        perror("sched_setaffinity()");
    i = 0;
    while (1) {
        i++;
        printf("%d\n", i);
    }
}
```


Elahe/e... ⌘1
parham... ⌘2
Elahe/S... ⌘3
parham... ⌘4
Introdu... ⌘5
Users/p... ⌘6
parham... ⌘7
Users/p... ⌘8
TA-Hom...
Users/p...
whil...
while:... ⌘9
Tasks: 320 total, 1 running
Load average: 2.06 1.92 1.81
Uptime: 3 days, 21:21:02
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
65209 parham 31 0 2405M 2884 0 R 2.0 0.0 0:00.00 htop
F1Help F2Setup F3SearchF4InvertF5Tree F6SortByF7Nice -F8Nice +F9Kill F10Quit


Elahe/e... ⌘1
parham... ⌘2
Elahe/S... ⌘3
parham... ⌘4
Introdu... ⌘5
Users/p... ⌘6
parham... ⌘7
Users/p... ⌘8
TA-Hom...
Users/p...
whil...
while:... ⌘9
Tasks: 323 total, 1 running
Load average: 2.11 1.94 1.81
Uptime: 3 days, 21:21:14
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
65209 parham 31 0 2405M 2892 0 R 1.0 0.0 0:00.00 htop
F1Help F2Setup F3SearchF4InvertF5Tree F6SortByF7Nice -F8Nice +F9Kill F10Quit



سوال ۷

در SCS مدیریت و زمان بندی threadها بر عهده کرنل میباشد و به این ترتیب هر یک یا چند thread سطح کاربر به تعداد kernel thread مرتبط میشود ولی در مدل PCS مدیریت این threadها بر عهده کاربر و کتابخانه‌ای در سطح کاربر هست.

سوال ۸

در سیستم‌های بی‌درنگ از آنجایی که نیاز هست در یک بازه زمانی کوتاه به یک event پاسخ داده شود نیاز هست که فاصله زمانی event تا زمان اجرای handler حداقل گردد.