

باسمه تعالی



دانشگاه صنعتی اصفهان
سیستم های عامل – تمرین اول

سوالات تئوری

سوال ۱: کد زیر قسمتی از برنامه manyForks.c را نشان می دهد.

```
1 int main(){
2     int i, pid;
3     for(i=0;i<=2;i++){
4         fork();
5         printf("%d\n",getpid());
6     }
7     pid = wait();
8     printf("%d\n",pid);
9     return(0);
10 }
```

فرض کنید پروسس اولیه که توسط shell ایجاد می شود pid برابر با ۱۰۰ دارد و همچنین pid های مربوط به پروسسهای بعدی به ترتیب با افزایش یک واحدی بدست می آیند. نحوه عملکرد زمانبند را هم بدین صورت در نظر بگیرید: پس از اجرای هر فراخوانی سیستمی (هر fork ، wait و یا return) اجرا می شود و پروسسی را از بین پروسسهای آماده برای اجرا انتخاب می کند که pid کمتری دارد. الف) درخت پروسسها را ترسیم کنید. ب) آنچه در ترمینال چاپ می شود را بنویسید. ج) ترتیب به اتمام رسیدن پروسسها به چه نحو است؟

سوال ۲:

الف) تصور کنید شما می خواهید سیستم عاملی را بر روی یک پردازنده ایجاد کنید که تنها از وقفه ها (interrupt) و استثناها (exception) پشتیبانی می کند اما دستوری مرتبط با فراخوانی سیستمی (trap) ندارد (مثلا دستور int ندارد). آیا می توانید به روشی هوشمندانه توسط وقفه ها و یا استثناها، فراخوانی های سیستمی را نیز پیاده سازی نمایید؟

ب) تصور کنید شما می خواهید سیستم عاملی را بر روی یک پردازنده ایجاد کنید که تنها از فراخوانی های سیستمی و استثناها پشتیبانی می کند اما از وقفه پشتیبانی نمی کند. آیا می توانید به روشی هوشمندانه توسط فراخوانی های سیستمی و یا استثناها، وقفه ها را نیز پیاده سازی نمایید؟

سوال ۳: بهینه بودن زمانبند SJF را برای حالتی که کارهای محاسباتی صرف (cpu-bounded) تقریباً همزمان به سیستم وارد می‌شوند و مدت زمان مورد نیاز برای پردازش هر یک از آنها نیز معلوم است را اثبات نمایید.

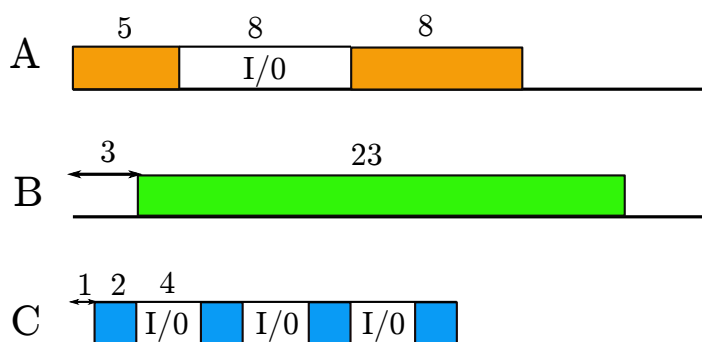
سوال ۴:

الف) زمانبند چرخشی (RR) را در نظر بگیرید. فرض کنید اتلاف زمان برای تعویض محیط (context switch) ناچیز است. در شرایطی که کوانتوم زمانی این زمانبند اجازه اجرای تنها یک دستور از هر کار را بدهد، عملکرد این زمانبند به کدامیک از زمانبندهای FIFO و SJF نزدیک‌تر است؟ توضیح دهید (فرض کنید کارها تقریباً همزمان به سیستم وارد می‌شوند).

ب) به سوال بخش الف با فرض اینکه کوانتوم زمانی به اندازه کافی بزرگ است که قبل از به اتمام رسیدن آن هر کاری که نوبت آن بوده پایان یابد پاسخ بدهید.

سوال ۵:

حجم کار (workload) زیر را در نظر بگیرید.



ترتیب اجرای کارها را به ازای زمانبندهای زیر را بدست آورید و زمانبندها را با معیارهای زمان پاسخ و زمان چرخش با هم مقایسه کنید.

الف) SJF

ب) PSJF

ج) RR با کوانتوم زمانی ۲

د) MLFQ با سه سطح و کوانتوم‌های زمانی ۲، ۴ و ۸ (در هر سطح کارها به صورت RR ساده زمانبندی می‌شوند)

سوالات تئوری – عملی

سوال ۱ مقایسه فراخوانی سیستمی (system call) با فراخوانی تابع (procedure call) :

الف) دستور int و call را با هم مقایسه کنید.

ب) دستور ret و iret را با هم مقایسه کنید.

ج) نحوه مدیریت stack در فراخوانی سیستمی و فراخوانی تابع چه تفاوتی با هم دارند؟ در هر مورد مشخص کنید با چه مقادیری مقداردهی اولیه می‌شوند و توسط نرم‌افزار و یا سخت‌افزار این کار صورت می‌پذیرد.

د) برنامه زیر را اجرا کنید و خروجی آن را تحلیل نمایید.

```
1#include <stdio.h>
2#include <time.h>
3#include <unistd.h>
4#define LOOP 10000
5void proc(void)
6{
7    return;
8}
9int main(int argc, char const *argv[])
10{
11    clock_t start, end;
12    double cpu_time_function, cpu_time_syscall;
13
14    start = clock();
15    for (int i = 0; i < LOOP; ++i)
16        proc();
17    end = clock();
18    cpu_time_function = ((double) (end - start)) / CLOCKS_PER_SEC;
19
20    start = clock();
21    for (int i = 0; i < LOOP; ++i)
22        getpid();
23    end = clock();
24    cpu_time_syscall = ((double) (end - start)) / CLOCKS_PER_SEC;
25
26    printf("Procedure call (%d times): %f sec\n", LOOP, cpu_time_function);
27    printf("System call  (%d times): %f sec\n", LOOP, cpu_time_syscall);
28}
```

i راهنمایی: برای پاسخ به بخش‌های الف و ب می‌توانید به مرجع زیر بخش 3.7 مراجعه فرمایید:

Bryant, R. E., David Richard, O. H., & David Richard, O. H. (2003). Computer systems: a programmer's perspective. Upper Saddle River: Prentice Hall.