به نام خدا

تمرین سری چهارم سیستم‌عامل

توضیحات:

- پاسخ به تمرین‌ها باید به صورت انفرادی صورت گیرد و در صورت مشاهده‌ی هر گونه تقلب نمره‌ی صفر برای کل تمرین‌ها منظور خواهد شد.
- تمیزی و خوانایی جواب تمرین‌ها از اهمیت بالایی برخوردار است. در صورت ناخوانایی جواب‌ها پس از تذکر برای بار اول، نمره‌ای به تمرین‌ها داده نخواهد شد.
- لطفا جواب تمرین‌ها را در قالب یک فایل PDF با نام "HW … _StudentNumber.pdf" در سایت درس و در مهلت معین شده بارگزاری نمایید.
- در صورت داشتن اشکال می‌توانید از طریق ایمیل درس os.1401fall@gmail.com با تدریسیاران درس در ارتباط باشید.

نیم‌سال اول ۰۱-۰۲

سوال ۱) در خصوص انواع فرایندها به سوالات زیر پاسخ دهید.

الف) فرایند فرزند چگونه منابع مورد نیاز خود را تامین می‌کند؟ آیا می‌تواند از منابع والد استفاده کند؟

ب) همانطور که می‌دانید فرایند فرزند ممکن پیش از اتمام اجرا، توسط فرایند والد به پایان برسد. توضیح دهید که فرایند والد به چه دلایلی ممکن است تصمیم بگیرد فرایند فرزند پایان یابد؟

ج) به کد زیر دقت کنید. آیا قطعه مربوط به والد اجرا می‌شود؟ توضیح دهید.

```c
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main() {
    pid_t pid;
    /* fork a child process */
    pid = fork();

    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0) { /* child process */
        execlp("/bin/ls","ls",NULL);
        printf("Child process");
    }
    else { /* parent process */
        wait(NULL);
        printf("Child Complete");
        exit(0);
    }
}
```

سوال ۲) به سوالات زیر پاسخ دهید

- دستورالعمل TestAndset یک CPU را شرح دهید. به طور معمول از چه نوع blocking استفاده می شود؟ چه زمانی این روش مناسب است؟
- چهار روش را که میتوانیم با آن‌ها با مشکل deadlock روبرو بشویم را به طور مختصر توضیح دهید و بگویید هر روش برای چه زمانی مناسب است.
- اگر در مسئله غذا خوردن فیلسوف‌ها یک chopstick ششم در وسط میز قرار دهیم آیا مشکل deadlock را حل کرده‌ایم؟ اگر جوابتان مثبت است بگویید که کدام شرط را حذف کرده‌ایم در غیر این صورت توضیح دهید چرا حل نشده است.

سوال ۳) الگوریتم Peterson را برای پشتیبانی از N پردازه بازنویسی کنید و شروط انحصار متقابل، پیشرفت و انتظار محدود را در الگوریتمتان بررسی کنید.

سوال ۴) برنامه زیر را که از Pthreads API استفاده می‌کند. خروجی برنامه در خط P چیست؟ تمامی فراخوانی‌های سیستمی با موفقیت اجرا می‌شوند.

```
#include <pthread.h>
#include <stdio.h>

int value = 0;
void *runner(void *param); /* the thread */

int main(int argc, char *argv[])
{
pid_t pid;
pthread_t tid;
pthread_attr_t attr;

   pid = fork();

   if (pid == 0) { /* child process */
      pthread_attr_init(&attr);
      pthread_create(&tid,&attr,runner,NULL);
      pthread_join(tid,NULL);
      printf("CHILD: value = %d",value); /* LINE C */
   }
   else if (pid > 0) { /* parent process */
      wait(NULL);
      printf("PARENT: value = %d",value); /* LINE P */
   }
}

void *runner(void *param) {
   value = 5;
   pthread_exit(0);
}
```