



تحویل در روز جمعه مورخ ۱۳۹۶/۰۸/۵ ساعت ۲۳:۵۵ از طریق سایت درس

### لطفا نکات زیر را در نظر بگیرید

- ۱- سوال ها به صورت واضح و بدون حاشیه های اضافی پاسخ بدهید.
- ۲- پاسخ های خود را به صورت خوانا (یا تایپ شده) بنویسید.
- ۳- پاسخ های خود را با کیفیت مناسب و خوانا اسکن کرده و به صورت یک فایل pdf در آورید.
- ۴- عکس های اسکن شده را قبل از تبدیل به pdf در جهت مناسب (rotation) قرار بدهید.
- ۵- تمامی فایل های مربوط به پاسخ خود را درون یک پوشه قرار داده، نام پوشه را به صورت name\_studentID قرار بدهید و آن را فشرده کنید.

۱. دو الگوریتم زمان بندی (Scheduling) زیر موجود است:  
(a) الگوریتم A جواب بهینه (بهترین جواب موجود) را پیدا می کند ولی زمان اجرای آن بسیار زیاد است.  
(b) الگوریتم B جواب خوب اما ضعیف تر از قسمت a پیدا می کند ولی زمان اجرای آن بسیار کم است.  
بنویسید که هر کدام در چه دسته Application هایی کاربرد دارند؟
۲. تفاوت Context switch و Swapping را بنویسید؟
۳. کدام یک از مراحل Context switch بصورت نرم افزاری و کدام یک بصورت سخت افزاری انجام می شوند؟
۴. در چه سناریوهایی زمانی که سیستم عامل صرف Context switching می کند هیچ تاثیری در کارایی سیستم ندارد؟
۵. در هر یک از حالات زیر مزایا و معایب را بیان کنید.

حالت ۱	حالت ۲
Parent and child share all resources and execute concurrently	Parent and child don't share all resources but execute concurrently
حالت ۳	حالت ۴
Parent and child share all resources and don't execute concurrently	Parent and child don't share all resources and don't execute concurrently



---

تحویل در روز جمعه مورخ ۱۳۹۶/۰۸/۵ ساعت ۲۳:۵۵ از طریق سایت درس

---

۶. فرض کنید کد زیر موجود است و هیچ خطایی (error) از اجرای کامل فرآیند (process) جلوگیری

نمی‌کند و فرآیند با سیگنال با وضعیت صفر پایان می‌یابد. به سوالات زیر پاسخ دهید:

(a) حداکثر تعداد دفعاتی که عبارت "SIGKILL received" چاپ (Print) می‌شود را بیان کنید.

(b) لیست کلیه مقادیر متغیر count که چاپ (Print) می‌شود را بیان کنید.

```
int count = 0;
void killhandler(int sig)
{
    printf("SIGKILL received\n");
    return;
}
void childhandler(int sig)
{
    int status;
    wait(&status);
    count += WEXITSTATUS(status);
    return;
}
main()
{
    int i; // for loop iterator
    pid_t pid[5]; // pids of child processes
    Signal(SIGKILL, killhandler);
    Signal(SIGCHLD, childhandler);
    //Fork 3 child processes
    for(i=0; i<5; i++)
    {
        pid[i] = fork();
        if(!pid[i]){ // If child process
            Signal(SIGKILL, SIG_DFL);
            exit(5);
        }
    }
    //Parent process only
    for(i=0; i<5; i++)
    {
        kill(pid[i], SIGKILL);
    }
    sleep(5);
    printf("count = %d\n", count);
    exit(0);
}
```



---

تحویل در روز جمعه مورخ ۱۳۹۶/۰۸/۵ ساعت ۲۳:۵۵ از طریق سایت درس

---

۷. قطعه کد زیر که به زبان C نوشته شده است را در نظر بگیرید. خروجی آنرا بیان کنید.

```
#include <stdlib.h>
#define NUM_FORKS 6
char array[NUM_FORKS+2]
int pos = 0;
char outs[9] = {'1','1','8','5','2','2','4','1','3'};
void work(void* id)
{
    int index = (((int*)id))*2;
    char writeMe = outs[index];
    array[pos++] = writeMe;
}
int main()
{
    char three = '3';
    int i;
    int pid[NUM_FORKS];
    for(i = 0; i<NUM_FORKS; i++)
    {
        if(!(pid[i] = fork() )){
            work((void*)&i);
            exit(0);
        }
        waitpid(pid[i], NULL, 0);
    }
    array[pos++] = three;
    array[pos] = '\0';
    printf("%s", array);
    exit(0);
}
```