

به نام خدا



درس سیستم های عامل

---

تمرین اول تئوری

---

مدرس : دکتر انتظاری

طراح : سارا سادات یونسی

## قوانین

دانشجویان محترم لطفاً نکات تکمیلی زیر را در تمامی تمرینها در نظر بگیرید .

- در مجموع تمام تمرینها، ۲۴۰ ساعت تأخیر در ارسال پاسخها مجاز است به همین جهت زمان اعالم شده به هیچ وجه قابل تمدید نیست و در صورت نیاز میتوانید از این زمان استفاده کنید، با پایان یافتن این زمان نمره شما ۰ لحاظ خواهد شد .
- پروژه های درسی مهلت تأخیر ندارند .
- انجام تمرین و کوییز به صورت یکنفره میباشد.
- فرمت ارسالی تمرین تئوری حتماً باید بهصورت latex یا word باشد.
- در صورت مشورت کردن در مورد سؤال خاصی ذکر اسامی همراه با سؤال موردنظر، در گزارش کار طرفین الزامی هست غیر این صورت مشاهده هرگونه تشابه در گزارش کار یا کدهای پیادهسازی، به منزله تقلب برای طرفین در نظر گرفته میشود.
- تمامی منابع استفاده شده بهصورت دقیق ذکر شوند . همچنین استفاده از کدهای آماده بدون ذکر منبع و بدون تغییر به منزله تقلب خواهد بود و نمره تمرین شما صفر در نظر گرفته میشود .
- بخش زیادی از نمره شما مربوط به گزارش کار و روند حل مسئله است. لطفاً به موارد خواسته شده به صورت کامل، دقیق و بدون ابهام پاسخ دهید .
- لطفاً گزارش، فایل کدها و سایر ضامائم موردنیاز را با فرمت زیر در سامانه مدیریت دروس بارگذاری نماید به جای n شماره تمرین قرار داده میشود HWn\_[studentNumber].zip .
- درصد تشابه بالای ۳۰ درصد از دید quera، تقلب لحاظ میشود و بعد از تکرار برای بار دوم، نمره این تمرین و تمرین قبلی که تقلب آن گرفته نشده، ۰ لحاظ خواهد شد .
- زمان ارسال کوییزهای کلاس حل تمرین تا ساعت ۱۲ همان شب خواهد بود و این مورد به هیچ وجه تمدید نخواهد شد.
- فرض کوییزهای کلاس حل تمرین این است که این امتحان بهصورت book open برگزار نمیشود و اجازه استفاده از کدهای آماده در سایت ها را نخواهید داشت.

۱. برنامه ی زیر را تحلیل کنید و بگویید خروجی آن چیست :

```
1. #include <sys/types.h>
2. #include <sys/types.h>
3. #include <stdio.h>
4. #include <unistd.h>
5. int main(int argc, char *argv[]) {
6.
7.     pid_t pid = fork();
8.
9.     if (pid < 0) {
10.         fprintf(stderr, "Fork Failed");
11.         return 1;
12.     }
13.
14.     else if (pid == 0) {
15.         printf("I'm the child \n");
16.     }
17.
18.     else {
19.         wait(NULL);
20.         printf("Child Complete \n");
21.     }
```

۲. در برنامه ی زیر، چند process در انتها تولید می شود؟ توضیح دهید و tree process مربوطه را بکشید.

```
int funcfork(){
    fork();
    fork();

    printf("Who are you?!\n");
}
int main() {
    funcfork ();
    printf("Who are you?!\n");
    exit(0);
}
```

۳. ساختار Symmetric multiprocessing را با رسم نمایی از آن شرح دهید .

۴. برای به اشتراک گذاشتن اطلاعات، می توان از passing message یا memory shared استفاده کرد. خوبی ها و بدی های هر کدام را بنویسید و بگوئید که از هر کدام بهتر است در چه شرایطی استفاده کنیم.

۵. اگر در یک وب سرور multithreaded تنها راه خواندن (read) از یک فایل، فراخوانی سیستمی مسدود کننده معمولی read() باشد، به نظر شما از user-level threads یا kernel-level threads هسته استفاده می شود؟ چرا؟

۶. چرا یک thread با فراخوانی thread\_yield() به طور داوطلبانه به CPU خود را ارائه می دهد؟ از آنجایی که هیچ وقفه دوره ای وجود ندارد و ممکن است هرگز CPU را برنگرداند.

۷. هنگامی که یک فرآیند با استفاده از عملیات fork() یک فرآیند جدید ایجاد می کند، کدام یک از حالت های زیر بین فرآیند والد و فرآیند فرزند به اشتراک گذاشته می شود؟ چرا؟

a. Stack

b. Heap

c. Shared memory segments

۸. اگر  $f_E$  و  $f_I$  را بدانیم، از معادله Speedup برای تعیین  $S$  استفاده می کنیم.

اگر یک برنامه ۴۰ درصد کد خود را بهبود دهد بنابراین ( $f_E = 0.4$ ) تا ۲,۳ برابر سریعتر اجرا شود بنابراین ( $f_I = 2.3$ ) سرعت کلی سیستم  $S$  چیست؟

$$S = ( (1 - f_E) + (f_E / f_I) )^{-1}$$

$$\text{Speedup} = \frac{1}{(1 - \text{fraction enhanced}) + (\text{fraction enhanced} / \text{factor of improvement})}$$

۹. concurrent processing ها باعث به وجود آمدن سرشار و پیچیدگی اضافی در سیستم های عامل می شوند.

سه مورد از این سرشار ها را نام ببرید: