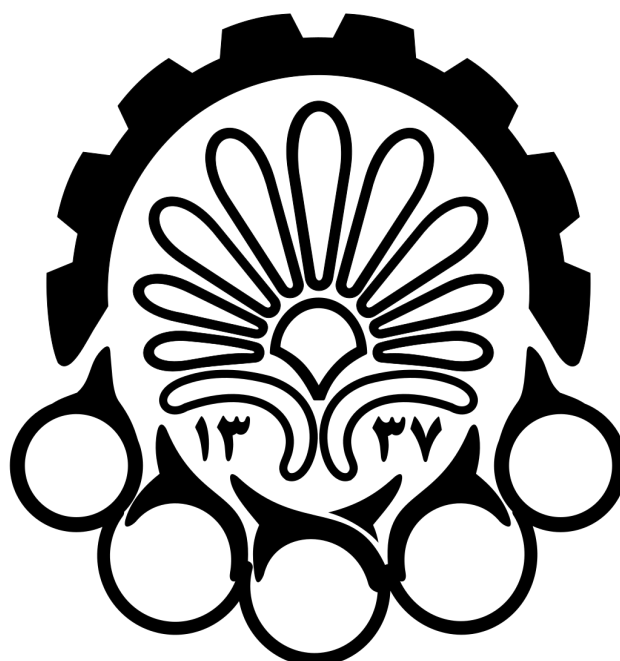


به نام خدا



**دانشگاه صنعتی امیرکبیر**  
**( پلی تکنیک تهران )**

سیستم‌های عامل - گروه ۱ (نیم‌سال دوم ۱۳۹۹-۱۴۰۰)

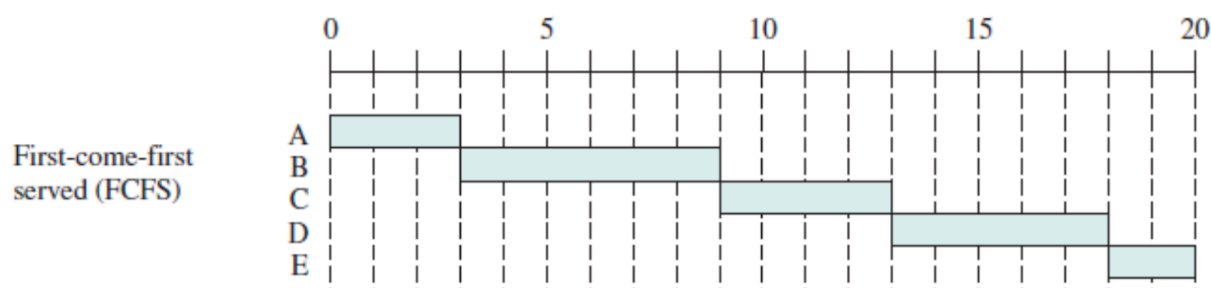
تمرین شماره ۳  
زمان‌بندی و همگام‌سازی  
(Scheduling & Synchornization)  
آخرین تاریخ بارگذاری پاسخ در courses:  
ساعت ۲۳:۵۹ روز ۲۱ خرداد ۱۴۰۰

## سوال ۱

سیستمی با شرایط زیر را در نظر بگیرید:

Process	Burst Time	Priority	Arrival Time
P1	50 ms	4	0 ms
P2	20 ms	1	20 ms
P3	100 ms	3	40 ms
P4	40 ms	2	60 ms

زمانبندی پردازنده را با استفاده از الگوریتم‌های FCFS، SJF، non-preemptive priority و RR (با کوانتوم زمانی ۳۰ میلی ثانیه) را همانند مثال زیر ترسیم کنید.



## سوال ۲

مجموعه فرآیندهای زیر را با مدت زمان پردازنده (CPU burst time) بر حسب میلی ثانیه در نظر بگیرید:

Process	Burst Time	Priority
$P_1$	2	2
$P_2$	1	1
$P_3$	8	4
$P_4$	4	2
$P_5$	5	3

فرض میشود فرآیندها در لحظه ۳ به ترتیب  $P_1$ ،  $P_2$ ،  $P_3$ ،  $P_4$  و  $P_5$  وارد میشوند. الگوریتمهای non-preemptive priority، SJF، FCFS (شماره بزرگتر، نشان از اولویت بالاتر است) و نوبت چرخشی، را در نظر بگیرید:

الف) زمان رفت و برگشت (Turnaround time) هر فرآیند با هر الگوریتم زمانبندی فوق چقدر است؟

ب) زمان انتظار (Waiting time) هر فرآیند با هر الگوریتم زمانبندی فوق چقدر است؟

ج) کدام یک از الگوریتمهای زمانبندی، میانگین زمان انتظار کمینه (برای کل فرآیندها) را دارد؟

### سوال ۳

نشان دهید اگر عملیات (wait) و (signal) سمافور به صورت اتمیک اجرا نشوند، در این صورت امکان نقض انحصار متقابل وجود دارد.

### سوال ۴

در راه حل زیر برای مسئله فیلسوفان غذاخور، فرض کنید روالهای برداشتن دو چنگال و گذاشتن هر چنگال از ابتدا تا انتها با رعایت کامل انحصار متقابل (Mutual Exclusion) انجام می‌شود. روال `take_forks(i)` دو چنگال راست و چپ را به ترتیب بررسی می‌کند و اگر هر دو موجود بودند بر می‌دارد، در غیر این صورت عمل بررسی تکرار می‌شود. روال `put_fork(i)` چنگال شماره `i` را می‌گذارد و از روال خارج می‌شود.

الف) بررسی نمایید آیا ممکن است بن بست پیش بیاید؟ اگر بله، مثال بیاورید.

ب) بررسی نمایید آیا ممکن است قحطی پیش بیاید؟ اگر بله، مثال بیاورید.

```
void philosopher(int i) {  
    Think();  
    take_forks(i);  
    Eat();  
    put_forks(i);  
    put_forks((i+1)%n);  
}
```

## سوال ۵

فرض کنید یک نانوايي وجود دارد که در مقابل آن دو صف تشکیل شده است. در هر لحظه فقط یک نفر میتواند در مقابل نانوا قرار گرفته و از او نان بگیرد. با استفاده از سمافور ها راه حلی ارائه کنید که :  
بیش از یک نفر مقابل نانوا قرار نگیرد و ثانيا هیچ یک از دو صف دچار قحطی نشوند.

همچنین می توانید پیاده سازی های توابع signal و wait را مطابق زیر فرض کنید:

```
wait(semaphore *s) {  
    S->value--;  
    if (S->value < 0) {  
        add this process to S->list;  
        block();  
    }  
}
```

```
signal(semaphore *s) {  
    S->value++;  
    if (S->value <= 0) {  
        remove a process P from S->list;  
        wakeup(P);  
    }  
}
```

## سوال ۶

شروط انحصار متقابل، پیشرفت و انتظار محدود را برای الگوریتم زیر بررسی کنید و دلیل خود را بنویسید.

```
do {  
    flag[i] = true;  
    turn = j;  
    while (!flag[j] || turn == j);  
    critical section  
    flag[i] = false;  
    remainder section  
} while (true);
```

## سوال ۷

انتظار مشغول (busy waiting) چیست؟ سایر انتظارهای موجود در سیستم عامل کدامند؟ آیا به صورت کلی می توان از انتظار مشغول اجتناب کرد؟ پاسخ خود را توضیح دهید.

## نحوه تحویل تمرین

پاسخ به سوالات را در قالب یک فایل پی دی اف (اسکن یا تایپ شده) با نام «HW3\_Student\_ID» در صفحه درس اپلود کنید. پاسخ‌های شما بایستی دقیق و خوانا باشند.

### جریمه دیرکرد

هر روز تاخیر در ارسال تمرین ۱۰٪ نمره منفی خواهد داشت. امکان اپلود تمرین تنها تا ۵ روز از تاریخ تعیین شده ممکن خواهد بود.

### جریمه تقلب

۱. همه دانشجویان بایستی که خود تمرین را انجام دهند و هرگونه تقلب یا ارسال کار دیگران یا کارهای موجود در وب که تمرین را به شکل جزئی یا کلی انجام داده است، غیرقابل پذیرش و عواقب شدیدی خواهد داشت.

۲. بنده و گروه حل تمرین تمام تلاش خود را برای شناسایی تقلب‌های احتمالی خواهیم کرد تا در نهایت یک ارزیابی عادلانه از همه دانشجویان عزیز داشته باشیم. ما از MOSS برای شناسایی فایل‌های مشابه (تمارین عملی) استفاده خواهیم کرد.

۳. در صورت شناسایی تقلبی که ۵۰ درصد یا پایینتر از کار را شامل می‌شود، دانشجوی مورد نظر اخطار اول را دریافت کرده و نمره «۰.۵- \*» بارم تمرین ۱» به ایشان تعلق می‌گیرد و در صورت شناسایی تقلبی که بیشتر از ۵۰ درصد کار را پوشش می‌دهد به دانشجوی مورد نظر اخطار دوم تعلق گرفته و نمره «۱- \*» بارم تمرین ۱» به ایشان تعلق می‌گیرد. علاوه بر این نمره منفی، گرفتن دو اخطار، به معنی لحاظ شدن نمره منفی برابر با بارم همه تمرینات خواهد بود.

در نهایت، هر گونه سوال در مورد تمرین و بخش‌های آنها را **تنها و تنها** از طریق سایت درس و ایجاد مباحثه با عناوین مرتبط مطرح بفرمایید.

تندرست و موفق باشید

تیم درس سیستم‌های عامل