



دانشکده ی مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر

در صورت داشتن سوال در مورد این
تمرین، سوال خود را با موضوع تمرین
با ایمیل زیر در میان بگذارید:
osfall2020@gmail.com

تمرین هفتم درس سیستم عامل

مهلت تحویل ساعت ۵۹:۲۳ روز ۲۳ آبان ۹۹

تمرینات را انفرادی حل کرده و در سایت مودل (courses.aut.ac.ir) با
قالب زیر بارگذاری نمایید:

StudentID_Name_Last Name

۱- شروط انحصار متقابل، پیشرفت و انتظار محدود را برای الگوریتم‌های زیر بررسی کرده و دلیل خود را بنویسید.
(الف)

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

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

(ب)

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

```
do
    flag i true
    turn ← i
    while (flag[j] & turn=i);
    flag[i] = false
```

۲- بدون استفاده از قفل و تنها با استفاده از دستور *compare-and-swap* تابع زیر را به گونه‌ای کامل کنید که به صورت اتمی عملیات جمع را انجام دهد. منظور از عملیات جمع اضافه شدن مقدار v به حافظه‌ای است که p به آن اشاره دارد. سپس توضیح دهید که تضمینی برای انجام شدن این عملیات وجود دارد یا خیر.

```
int add(int *p, int v)
{
    // TODO
    return *p + v;
}
```

پیاده‌سازی *compare-and-swap* را به صورت زیر در نظر بگیرید:

```
bool compare_and_swap(int *p, int old, int new)
{
    if(*p != old)
        return false;
    *p = new;
    return true;
}
```

۳- می‌دانید که برای پیاده‌سازی توابع *acquire()* و *release()* در قفل *mutex* باید از دستورات سخت‌افزاری اتمی استفاده کرد. این کار را با استفاده از دستور *test-and-set* انجام دهید.