

## تمرین ۲

تینا صداقت ۰۴۴ ۹۳۳۱

از روش test and set lock (TSL) استفاده می کنیم. روش TSL به صورت زیر کار می کند:

مقدار اولیه lock، false است.

- تست کردن lock
- ست کردن lock

اگر مقدار return برابر false باشد lock را می توان به دست آورد.

اگر true باشد، کسی lock را هنوز دارد.

دستور TSL به صورت زیر پیاده می شود:

```
Boolean TSL (Boolean i){
```

```
if (i==false){
```

```
  i=true;
```

```
  return true;}
```

```
else
```

```
  return false;}
```

که به صورت زیر در برنامه ها از آن استفاده می شود.

```
1 repeat
2 while(TSL(lock)==false)
3 no-op;
4 critical section
5 Lock = FALSE;
6 remainder section
7 until FALSE
```

مشکل موجود: ممکن است حالتی به وجود بیاید که در هنگام آپدیت کردن مقدار تایمر، به دلیل آمدن interrupt، تایمر به درستی نمایش داده نشود. فرض کنید ساعت ۲۳:۵۹ باشد و تابع getDateime در حال اجرای دستوراتش باشد و وقتی دارد مقدار ساعت را از ۲۳ به ۰۰ تغییر می دهد interrupt رخ دهد و مقدار دقیقه همان ۵۹ بماند. واضح است که تایمر مقدار غلط ۰۰:۵۹ را به جای ۰۰:۰۰ برای مدتی نشان می دهد.

راه حل: استفاده از TSL. به این صورت که هر دو تابع که دارای قسمت critical section هستند را توسط test & set lock کنترل کنیم. (باید در هنگام اجرا تابع دیگر تغییری در برنامه ندهد) وقتی مقدار lock به سیستم بر می گردد تنها یک پروسس مقدار false را بازگشت می دهد و دیگری منتظر تغییر مقدار lock می ماند و انجام نمی شود.

```
void GetDateTime(DateTimeType * DT){  
while(TSL(lock)==false)  
no-op;  
  
DT->day = TimerVal.day;  
DT->hour = TimerVal.hour;  
DT->minute = TimerVal.minute;  
DT->second = TimerVal.second;  
Lock = FALSE;  
}
```

```
void DateTimeISR(void){  
while(TSL(lock)==false)  
no-op;  
TimerVal.second++;  
  
if (TimerVal.second > 59){  
    TimerVal.second = 0;  
    TimerVal.minute++;  
    if (TimerVal.minute > 59) {  
        TimerVal.minute = 0;  
        TimerVal.hour++;  
        if (TimerVal.hour > 23) {  
            TimerVal.hour = 0;  
            TimerVal.day++;  
            ... etc.  
        }  
    }  
Lock = FALSE;  
}
```

