



کوییز دوم / سیستم‌های عامل 4001_02_1



کوییز دوم

Started on	Monday, 22 Azar 1400, 5:04 PM
State	Finished
Completed on	Monday, 22 Azar 1400, 5:31 PM
Time taken	27 mins 35 secs
Grade	8.00 out of 10.00 (80%)

Question 1

Correct

Mark 2.00 out of 2.00

Flag question

استفاده از spin lock جهت جلوگیری از دسترسی همزمان به داده‌های کرنل، در هیچ شرایطی روش مناسبی نیست زیرا باعث busy waiting و کاهش بهره وری CPU میشود

Select one:

- ☐ True
- ☒ False ✓

The correct answer is 'False'.

Question 2

Correct

Mark 2.00 out of 2.00

Flag question

در مورد راه حل نرم افزاری زیر که جهت حل مسئله تاحیه بحرانی پیشنهاد شده است، کدام جمله یا جملات صحیح است. اولیه flag ها false است.

```
p0{
while(true){
    flag[0]=true;
    while(flag[1]); /*do nothing*/
    /*critical_section*/
    flag[0]=false;
    /* remainder_section*/
}
```

```
p1{
while(true){
    flag[1]=true;
    while(flag[0]); /*do nothing*/
    /*critical_section*/
    flag[1]=false;
    /* remainder_section*/
}
```

- ☒ a. امکان رخداد بن بست وجود دارد. ✓
- ☐ b. امکان رخداد گرسنگی بدین معنی که یکی از پروسسها نتواند وارد CS شود در حالیکه پروسس دیگر متناوباً میتواند وارد شود، وجود دارد.
- ☒ c. شرط انحصار متقابل برآورده شده ولی پیشرفت وجود ندارد. ✗
- ☐ d. شرط انحصار متقابل برقرار نشده است

پاسخ شما صحیح می باشد

The correct answer is: امکان رخداد بن بست وجود دارد

Question 3

Complete

Mark 4.00 out of 6.00

Flag question

یک اتوبوس با ظرفیت k مسافر هر بار، مسافرانی را از یک ایستگاه اتوبوس سوار می‌کند. هر وقت اتوبوس به ایستگاه میرسد، حداکثر به k مسافر منتظر اجازه سوارشدن میدهد و پس از آن راه می افتد. مسافران باید صبر کنند تا اتوبوس برسد و سوار شوند. مسافرانی که بعد از رسیدن اتوبوس به ایستگاه، به ایستگاه میرسند، نباید بتوانند سوار شوند و باید برای دفعه بعدی در ایستگاه منتظر بمانند. یک thread برای سوارشدن مسافران وجود دارد که هر بار که مسافری سوار میشود تابع savor را فراخوانی میکند و یک thread برای اتوبوس وجود دارد که هر بار بعد از اینکه تعداد موردنظر مسافران را سوار کرد، با فراخوانی harekat شروع به حرکت می‌کند. در ادامه سمارفور و متغیرهای لازم برای حل مسئله آمده است

```
semaphore mutex = 1
semaphore bus_arrived = 0
semaphore mosafes_savar = 0
int waiting_mosafes = 0
```

همچنین کد thread سوارشدن مسافران نیز به صورت زیر نوشته شده است. با استفاده از فقط همین سمارفورها و متغیرهایی که تعریف شده و با توجه به کد thread سوارشدن مسافران، کد حرکت اتوبوس را بنویسید.

```
lock(mutex)
waiting_mosafes ++
unlock(mutex)
lock(bus_arrived)
savar()
unlock(mosafes_savar)
```

فرض میکنیم تابع `savar`، متغیر `waiting_mosafer` را یک واحد کم میکند

```
lock(mutex) // مسافراتی که بعد از رسیدن اتوبوس به ایستگاه، به ایستگاه میرسند، نباید بتوانند سوار شوند
unlock(bus_arrived)
while(waiting_mosafer > 0 && mosafer_savar <= k)
    lock(mosafer_savar);
harekat();
lock(bus_arrived)
unlock(mutex);
```

```
lock(mutex)
m = min(waiting_mosafer, k)
for(i=0; i<m; i++){
    unlock(buss_arrived)
    lock(mosafer_savar)
    waiting_mosafer - -;
}
unlock(mutex)
harekat()
```

(اینجا `lock` و `unlock` منظور همان `wait` و `signal` است که از انتجایی که تعریفها سمافور بوده مشخص است.)

Comment:

[Finish review](#)