

長庚大學105學年度第一學期作業系統第三次小考

系級:

姓名:

學號:

1. (30%) Please (a) define “Race Condition” and (b) “Critical Section”.

Answer: (a) It is a situation where the outcome of the execution depends on the particular order of process scheduling.

(b) Each process has a segment of code, called a critical section, whose execution must be mutually exclusive.

2. (30%) There three processes

- $P_1: a * b \rightarrow a$
- $P_2: a + c \rightarrow a$
- $P_3: a + d \rightarrow a$

P_1 should run before P_2 and P_3 do. The access to valuable “a” must be protected. The order of P_2 and P_3 is arbitrary. We have three semaphores, and they are initialized as $S_1=0$, $S_2=0$, and $S_3=1$. Now, the code of P_1 is provided as follows:

```
a = a * b;  
signal(S1);  
signal(S2);
```

Please provide the code of P_2 and P_3 .

Answer:

Process P_2 :

```
wait(S1) ;  
wait(S3) ;  
a = a + c ;  
signal(S3) ;
```

Process P_3 :

```
wait(S2) ;  
wait(S3) ;  
a = a + d ;  
signal(S3) ;
```

3. Does the following solution work for protecting the critical sections? (40%)

Process P_i :

```
do {  
    flag[i]=true;  
    while (flag[j] && turn==j) ;  
    critical section  
    turn=j;  
    flag[i]=false;  
    remainder section  
} while (1);
```

Process P_j :

```
do {  
    flag[j]=true;  
    while (flag[i] && turn==i) ;  
    critical section  
    turn=i;  
    flag[j]=false;  
    remainder section  
} while (1);
```

Answer:

► No. 如果照以下方式執行，Mutual Exclusion的條件會被違反：

- 不失一般性，我們假設turn的初始值是i
- P_j 第一次開始執行時 P_i 尚未被執行過
 - 由於此時 P_i 還沒執行所以flag[i] 應為false
 - 所以 P_j 可以順利進入critical section
- 在 P_j 進入critical section的這段期間內 P_i 也接著開始執行
 - 由於turn的初始值是i
 - 所以 P_i 也可以順利進入critical section
- 這時候 P_j 和 P_i 同時在critical section裡 → 違反Mutual Exclusion