

2152118 史乃宝 E03 操作系统

1. ~~D~~ D 2. D 3. A 4. B 5. B 6. C
7. ~~D~~ D 8. B 9. D 10. B 11. B 12. D

13. ①同时只允许一个进程使用的软件和硬件资源。
②访问临界资源的程序段。

14. 等待当前进程访问结束,并发送消息。

15. ① P ② V

16. $-(m-1) \sim 1$ 或 $[-(m-1), 1]$

17. $S.value < 0$

18. 互斥

19. ① P 操作 ② V 操作。

20. 只有一个

21. 同步

22. ① P 操作 ② V 操作。

23. ① PV 操作是一组原语操作。用于实现进程互斥或进程同步。P 操作用于请求一个资源而 V 操作用于释放一个资源。

② $S.value > 0$ 指当前资源还有剩余,可供其他进程使用。

$S.value = 0$ 指当前所有进程已分配到 S 资源,且资源没有剩余。

$S.value < 0$ 指当前资源已分配完,还有 $|S.value|$ 个进程正在等待资源。

解：我们定义计数器 waiting 用于记录等待数。
mutex 作为信号量访问 waiting 资源并修改。

主程序：

操作员 void boss ()

```
main ( )  
{  
  cobegin ( )  
  {  
    boss ( )  
    customer ( )  
  }  
}
```

```
{ while (1) {  
  P(customer) (顾客来唤醒)  
  P(mutex)  
  if (waiting  
  waiting = waiting - 1;  
  V(mutex) → V(boss)  
  work ( )  
  V(boss) 通知有空闲  
}
```

顾客：

void customer ()

```
{ while (1) {
```

由上就完成了复印店的任务。

~~P(boss)~~

P(mutex)

```
if (waiting < 5) {
```

```
  waiting = waiting + 1;
```

```
  V(customer)
```

```
  V(mutex);
```

```
  P(boss)
```

```
  get-work ( )
```

```
}
```

```
else {
```

```
  V(mutex);  
}}
```


25. C 26. B 27. A 28. D 29. C 30. B

31. B 32. B

33. ①安全状态 ②不安全状态.

34. 请求保持条件.

35. ①死锁避免 ②死锁预防 ③死锁解除.

36. (1) 最大需求: 已分配: 需求:

① 0 0 1 2	0 0 1 2	0 0 0 0
② 2 7 5 0	2 0 0 0	0 7 5 0
③ 6 6 5 6	0 0 3 4	6 6 2 2
④ 4 3 5 6	2 3 5 4	2 0 0 2
⑤ 0 6 5 2	0 3 3 2	0 3 2 0

可知先完成①之后资源变为 2 1 1 2 可以完成④

之后资源变为 ~~4 4 4 4~~ 4 4 6 6 之后可以完成⑤

之后资源变为 4 7 9 8 之后可以完成②

之后资源变为 6 7 9 8 之后可完成③

∴是完成的, 完成顺序为 $P_1 \rightarrow P_4 \rightarrow P_5 \rightarrow P_2 \rightarrow P_3$

(2) 还是上面, 如果 P_3 请求 (0, 1, 0, 0)

最大需求:	已分配:	需求:	资源: 2 0 0 0
① 0 0 1 2	0 0 1 2	0 0 0 0	
② 2 7 5 0	2 0 0 0	0 7 5 0	
③ 6 6 5 6	0 1 3 4	6 5 2 2	
④ 4 3 5 6	2 3 5 4	2 0 0 2	
⑤ 0 6 5 2	0 3 3 2	0 3 2 0	

P_1 完成后为 (2 0 1 2) P_4 完成后为 (4 3 6 6) P_5 完成后为 (4 6 9 8)

此时 P_2, P_3 都无法分配完成, 系统不安全, 所以会拒绝.