長庚大學106學年度第一學期 作業系統 第三次小考

系級:　　　　　　　姓名:　　　　　　　學號:

1. (30%) There three processes

   - ◦ P_1: a * b → a
   - ◦ P_2: a + c → a
   - ◦ P_3: a + d → a

   P_1 should run before P_2 and P_3 do. The access to valuable "a" must be protected in a critical session. The order of P_2 and P_3 is arbitrary. We have only one semaphore, and it is initialized as S_1=0. Now, the code of P_1 is provided as follows:

   a = a * b;
   signal(S_1);

   Please provide the code of P_2 and P_3.

Answer:

Process P_2:
wait(S_1);
a = a + c;
signal(S_1);

ProcessP_3:
wait(S_1);
a = a + d;
signal(S_1);

2. (30%) There three processes

   - ◦ P_1: a * b → a
   - ◦ P_2: a + c → a
   - ◦ P_3: b + d → b

   The access to valuables "a" and "b" must be protected in critical sessions. The execution order of P_1, P_2 and P_3 is arbitrary. We have two semaphores, and they are initialized as S_1=1 and S_2=1. Now, the code of P_1 is provided as follows:

   wait(S_2);
   wati(S_1);
   a = a * b;
   signal(S_1);
   signal(S_2);

   Please provide the code of P_2 and P_3.

Answer:

Process P_2:
wait(S_1);
a = a + c;
signal(S_1);

ProcessP_3:
wait(S_2);
b = b + d;
signal(S_2);

3. (40%) For the bounded-buffer problem with consumers and producers, the code of consumers is provided as follows. Please provide the code of producer.

Consumer:

```
do {
        wait(full);    /* control buffer availability */
        wait(mutex);    /* mutual exclusion */
        remove an item from buffer to nextp;
        signal(mutex);
        signal(empty);    /* increase item counts */
        consume nextp;
} while (1);
```

Producer:

```
do {
        produce an item in nextp;
        Code Line 1;
        Code Line 2;
        add nextp to buffer;
        Code Line 3;
        Code Line 4;
} while (1);
```

Answer: Check the slides.