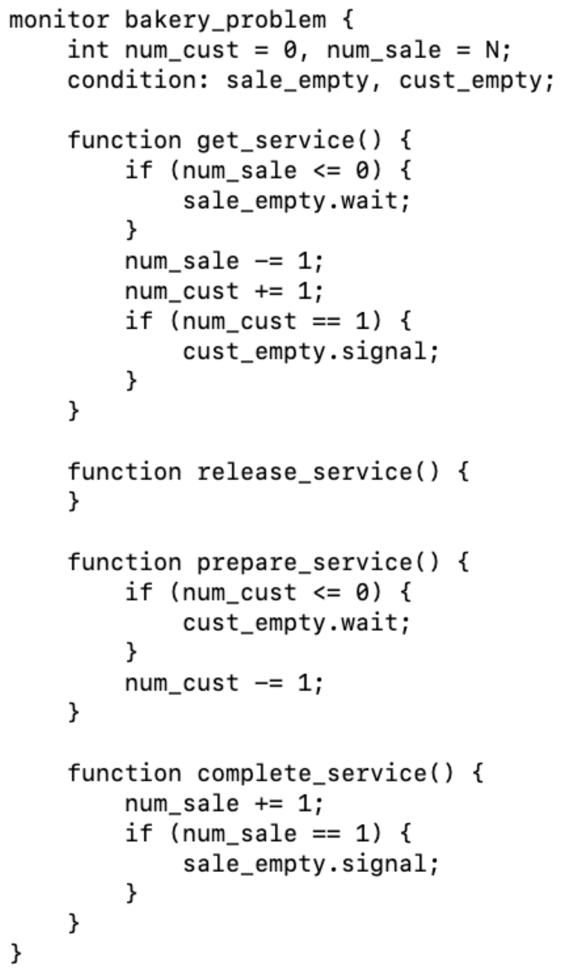
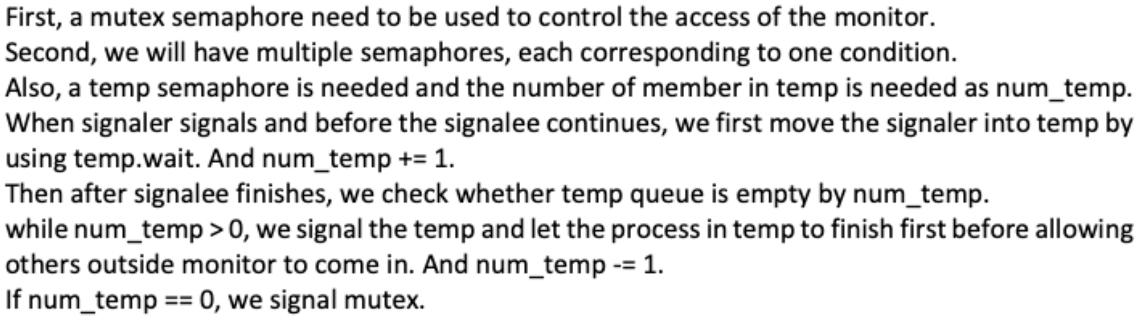
CS 4348/5348 Operating Systems, Homework #2

1.



```
monitor bakery_problem {
    int num_cust = 0, num_sale = N;
    condition: sale_empty, cust_empty;

    function get_service() {
        if (num_sale <= 0) {
            sale_empty.wait;
        }
        num_sale -= 1;
        num_cust += 1;
        if (num_cust == 1) {
            cust_empty.signal;
        }
    }

    function release_service() {
    }

    function prepare_service() {
        if (num_cust <= 0) {
            cust_empty.wait;
        }
        num_cust -= 1;
    }

    function complete_service() {
        num_sale += 1;
        if (num_sale == 1) {
            sale_empty.signal;
        }
    }
}
```

2.



First, a mutex semaphore need to be used to control the access of the monitor.
Second, we will have multiple semaphores, each corresponding to one condition.
Also, a temp semaphore is needed and the number of member in temp is needed as num_temp.
When signaler signals and before the signalee continues, we first move the signaler into temp by using temp.wait. And num_temp += 1.
Then after signalee finishes, we check whether temp queue is empty by num_temp.
while num_temp > 0, we signal the temp and let the process in temp to finish first before allowing others outside monitor to come in. And num_temp -= 1.
If num_temp == 0, we signal mutex.

3-

	Total	Allocated				MaxReq				Needed				Available
		P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	
R_1	3	1	0	0	0	1	1	2	2	0	1	2	2	2
R_2	6	1	1	0	1	2	2	2	1	1	1	2	0	3
R_3	4	0	0	1	1	2	1	1	2	2	1	0	1	2

	Requested			
	P_1	P_2	P_3	P_4
R_1	0	1	0	1
R_2	0	0	0	0
R_3	1	1	0	0

	Total	Allocated				MaxReq				Needed				Available
		P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	
R_1	3	1	1	0	1					0	0	2	1	0
R_2	6	1	1	0	1					1	1	2	0	3
R_3	4	1	1	1	1					1	0	0	1	0

	Needed			Available
After P2 continue	P1	P3	P4	
	0	2	1	1
	1	2	0	4
	1	0	1	1
After P1 continue		P3	P4	
		2	1	2
		2	0	5
		0	1	2
After P4 continue		P3		
		2		3
		2		6
		0		3
After P3 continue				3
				6
				4

All the process can finish, Grant the request.

4.

	Total	Allocated			Available	Requested		
		P_1	P_2	P_3		P_1	P_2	P_3
R_1	1	1	0	0	0	0	0	1
R_2	1	0	1	0	0	1	0	0
R_3	1	0	0	0	1	0	1	1
R_4	1	0	0	1	0	1	0	0

	Available	Requested		
		P_1	P_2	P_3
After P2 contimue :	0	0		1
	1	1		0
	1	0		1
	0	1		0

Dead Lock is detected. P1 and P3 are involved.

5.

(a). J1: 0x08000000, 0x08dfffff

J2: 0x08e00000, 0x0a1fffff

J3: 0x30000000, 0x39ffffff

J4: 0x3a000000, 0x3dffffff

(b). Physical address = 0x3a000000 + 0x04213a0C

= 0x3e21a0c > 0x3dffffff

access violation

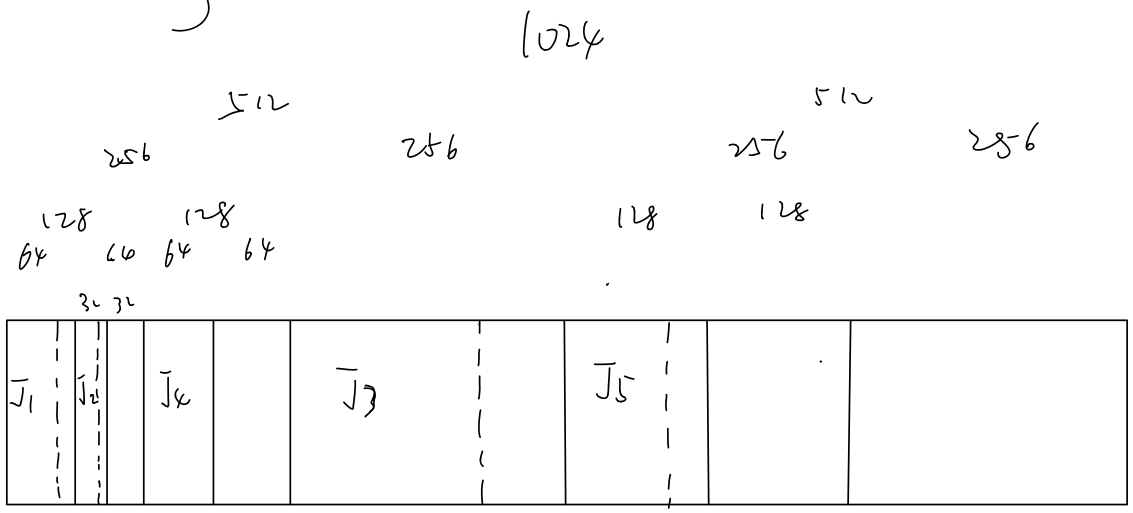
(c). Physical address = 0x3a000000 + 0x02f3f12c

= 0x3cf3f12c < 0x3dffffff

no access violation

6.

Memory allocation:



(a). J1: 0x00000000

J2: 0x04000000

J3: 0x10000000

J4: 0x08000000

J5: 0x20000000.

(b). (64 − 40) + (32 − 20) + (256 − 160) + (128 − 100)

= 24 + 12 + 96 + 28

= 160 MB

7.

a). 256MB = $2^8 \cdot 2^{20} = 2^{28}$ => 28 bit is enough.

b). 16KB = $2^4 \cdot 2^{10} = 2^{14}$ => 14 bit for offset

c). 16KB − 1

d). 0x0000ef5b => page# = 3 => frame# = 3

No access violation

e). 0x00017f5b => page# = 5 => access violation

8.

$$\text{Average access time} = 15 \times 90\% + 10\% \times 99.99\% \times (150 + 15) + 10\% \times 0.01\% \times (1 \times 10^6 + 150 + 15)$$

$$= 15 + 0.1 \times 150 + 0.00001 \times 1 \times 10^6.$$

$$= 40\ \text{ns}$$

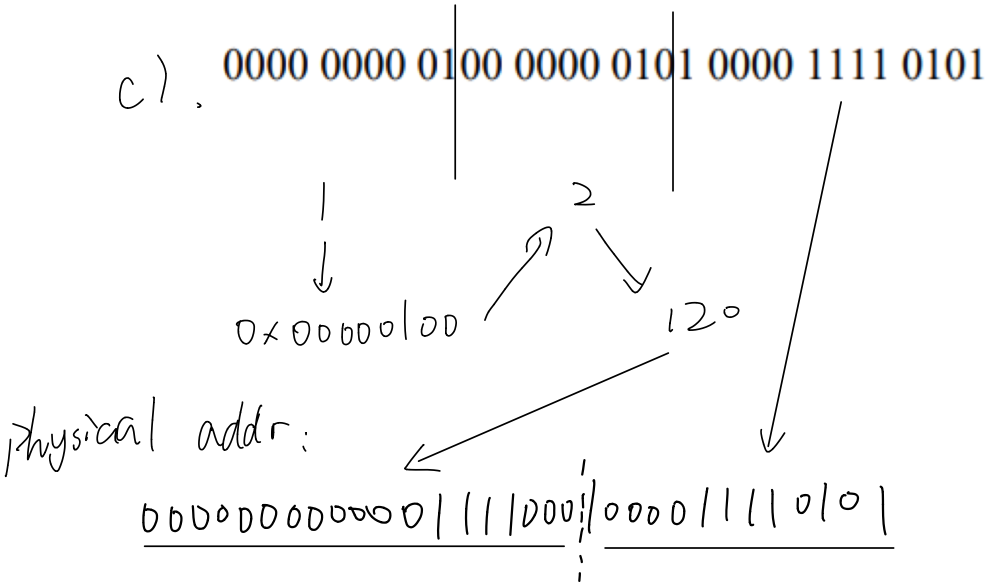
9.

a). $1K = 2^{10} \Rightarrow 10$ bits

b). page offset $= 8KB = 2^{13} \Rightarrow 13$ bit.

Ans $= 32 - 13 - 10 = 9$ bit.

c).



0000 0000 0100 0000 0101 0000 1111 0101

physical addr:

0000000000001111000¦1000011110101