Kamesh Vedula 68149804

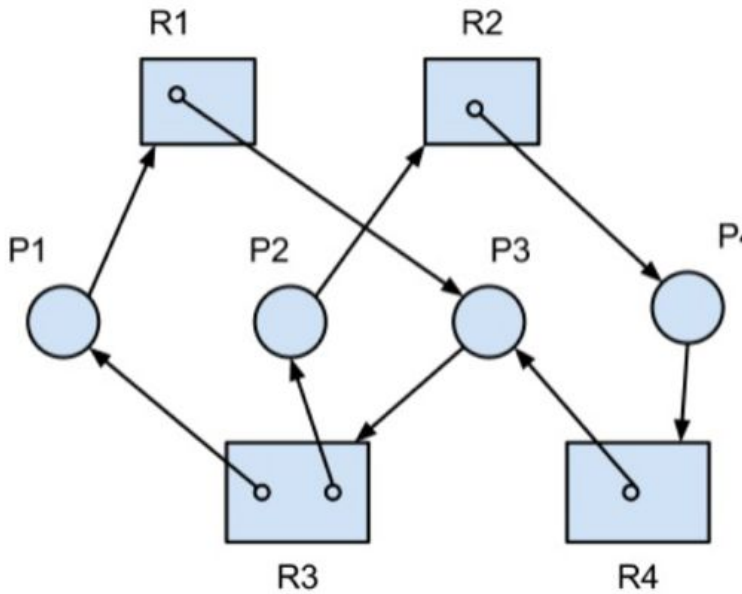
# ICS 143: Principles of Operating Systems

**Due Date: 11:55 pm, May 27, 2018 via Canvas**

**Homework #3 (Total Marks=100)**

## Question 1. Deadlocks (20 pts)

Consider the following resource allocation graph:




Cycles:

- P1 → R1 → P3 → R3 → P1
- P1 → R1 → P3 → P2 → R2 → P4 → R4 → P3 → R3 → P4

A. Determine if there is a deadlock.

If yes, indicate the processes and resources involved. Show how the deadlock can be resolved through addition of resources.

If not, argue why this is the case, i.e. there is no deadlock. In either case, provide a feasible sequence of processes to show completion. (10 pts)

Yes, there does exist a deadlock. All the resources can be reached by one process at a time, allowing for mutual exclusion. As well, to confirm the rest of the conditions to acheive a deadlock (Hold & wait, Circular wait, and No Preemption), we see that each process has some allocated resources and some request for other resources. As well they form a circular chain, confirming the deadlock.

To resolve this, we can add resources, like another instance to R4. This will allow P4 to be executed, thus releasing R2 for P2. Then P2 will release R3 for P3 and P3 will release R4 & R1 for P1, ultimately allowing P1 to finish execution.

B. Consider a system having $p$ processes, where each process needs a maximum of $m$ instances of resource type R1. Given that there are $r$ instances of resource type R1 in total, what is the minimum value of $r$ as a function of $p$ and $m$ to ensure that the system is deadlock-free? (10 pts)

$r = p(m-1)+1$

# Question 2. Safe state (20 pts)

Consider the following snapshot of a system:

| ***Process*** | ***Allocation*** | | | | ***Max*** | | | | ***Available*** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **A** | **B** | **C** | **D** | **A** | **B** | **C** | **D** | **A** | **B** | **C** | **D** |
| **P0** | 3 | 0 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 0 | 2 | 0 |
| **P1** | 0 | 1 | 0 | 0 | 0 | 2 | 1 | 2 | | | | |
| **P2** | 1 | 1 | 1 | 0 | 4 | 2 | 1 | 0 | | | | |
| **P3** | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | | | |
| **P4** | 0 | 0 | 0 | 0 | 2 | 1 | 1 | 0 | | | | |

a) What is the content of the matrix ***Need***? (8 pts)

| ***Process*** | ***Need*** | | | |
|---|---|---|---|---|
| | A | B | C | D |
| P0 | 1 | 1 | 0 | 0 |
| P1 | 0 | 1 | 1 | 2 |
| P2 | 3 | 1 | 0 | 0 |
| P3 | 0 | 0 | 1 | 0 |
| P4 | 2 | 1 | 1 | 0 |

Need[i,j] = Max[i,j] − Alloc[i,j]

b) Is the system in a safe state? If yes, give a safe sequence of processes. If not, explain why the system is not in a safe state. (6 pts)

yes, it is in a safe state. Possible sequence of processes is:

P3 → P4 → P0 → P2 → P1

c) If a request from process P1 arrives for (0,0,1,0), can the request be granted immediately? Please state the reason. If after this, another request from process P4 arrives for (0,0,1,0), can this request be granted immediately? Please state the reason. (6 pts)

if P1 requests (0,0,1,0) then it can be granted immediately as there is already a possible execution order which keep the system in a safe state. But, if P4 also requests (0,0,1,0) right after the system won't be in a safe state anymore since there will be a deadlock, thus not granting P4's request immediately.

## Question 3. Memory management (20 pts)

a) What is the main difference between external and internal fragmentation? (5 pts)

Internal Fragmentation: when a request uses a block of memory, but does not use the entire block, leaving free blocks of memory.

External Fragmentation: when there is a memory block for the request, but the block is not contiguous.

b) Consider a system that uses a fixed-partition scheme, with equal partitions of size $2^{12}$ bytes, and the main memory has $2^{32}$ bytes. A process table is maintained with a pointer to the resident partition for each resident process. How many bits are required for the pointer in the process table? Show all your steps. (6 pts)

Size of equal size partitions in fixed partitioning scheme = $2^{12}$ bytes

Size of main memory = $2^{32}$ bytes

# of bits required for pointer:

can be found by finding out # of partitions needed.

$$\# \text{ of partitions} = \frac{\#\text{ of bytes in main memory}}{\#\text{ of bytes in each partition}} = \frac{2^{32}}{2^{12}} = 2^{20}$$

$\Rightarrow$ Thus, # of bits required is 20 bits.

c) Consider a system that uses the variable-partition scheme. Given five memory partitions of 100KB, 500KB, 200KB, 300KB and 600KB in order, how would the first-fit, best-fit and worst-fit algorithm work with incoming processes of 221KB, 471KB, 127KB, and 451KB? (9 pts)

First Fit:

221 KB → 500 KB partition

471 KB → 600 KB partition

127 KB → new partion 500 KB − 221 KB = 279 KB Partition

451 KB has to wait,

Best Fit:

221 KB → 300 KB partition

471 KB → 500 KB partition

127 KB → 200 KB partition

451 KB → 600 KB partition

Worst Fit:

221 KB → 600 KB partition

471 KB → 500 KB partition

127 KB → 600 KB − 221 KB = 379 KB partition

451 KB has to wait.

## Question 4: Paging and TLB [20 Pts]

A computer keeps its page tables in memory. The overhead required for reading a value from memory is 3 microseconds. The computer has a TLB, which holds pairs of virtual pages and physical page frames: <virtual page, physical page frame>.

a) What is the effective access time if the TLB has an access time of 500 ns, and we observe a 20% TLB miss rate? Show your work. (10 pts)

3 ms = 3000ns

When page # in TLB, takes 500ns to search TLB & 3000 ns to access memory = 3500ns

If failed to find in TLB (500ns), then we need to access memory (3000ns), then access desired byte in memory (3000ns) = 6500 ns

hit miss

effective access time = $\underbrace{0.80 \times 3500}_{2800} + \underbrace{0.2 \times 6500}_{1300}$ = 4100ns

b) What is the effective access time if the TLB has an access time of 900 ns, and we observe a 95% hit rate? Show your work. (10 pts)

3 ms = 3000ns

When page # in TLB, takes 900 ns to search TLB & 3000 ns to access memory = 3900ns

If failed to find in TLB (900ns), then we need to access memory (3000ns), then access desired byte in memory (3000ns) = 6900ns

hit miss

effective access time = $\underbrace{0.95 \times 3900}_{3705} + \underbrace{0.05 \times 6900}_{345}$ = 4050ns

## Question 5: Memory Management[20 pts]

Consider the following segment table:

| Segment | Base | Length |
|---|---|---|
| 0 | 255 | 400 |
| 1 | 1924 | 30 |
| 2 | 3313 | 527 |
| 3 | 115 | 80 |
| 4 | 2890 | 125 |

What are the physical addresses for the following logical addresses formatted as **<segment, offset>?**

a) **[4 pts]** 0, 230

255 + 230 = 485

b) **[4 pts]** 1, 40

illegal reference, trap

c) **[4 pts]** 2, 500

3313 + 500 = 3813

d) **[4 pts]** 3, 400

illegal reference, trap

e) **[4 pts]** 4, 112

2890 + 112 = 3002