# **Luke Pepin - Homework 3**

# **Released: March 21, 2024**

# **Due: March 27, 2024, 11:59pm**

**Problem 1 (5 points):** How is a virtual address in MULTICS converted to physical address?

The translation from virtual address to physical address in MULTICS is typically done through the use of segmentation and paging. Segmentation divides the virtual address space into segments, while paging translates segment addresses into physical addresses. The mapping of virtual to physical addresses is managed by these two ensuring efficient memory access.

**Problem 2 (5 points):** Assume that there are 2048 entries in the segment table, and there can be at most 128 pages in a segment. If the virtual address is 32 bits, what is the maximum size of a page you can have?

2048 Entries in segment table (11 bits for the segment number)

Maximum of 128 pages in a segment (7 bits for the page number)

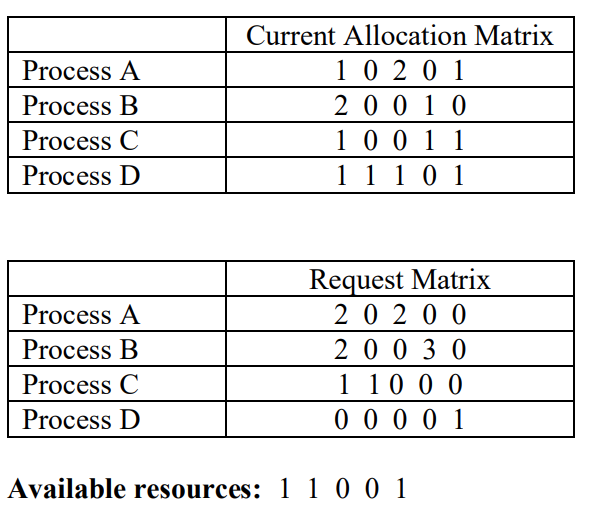
Virtual address size is 32 bits

Offset = Total – bits for segment – bits for page

Offset = 32-11-7 = 14 Bits

**2^14 = 16384 Bytes** is the maximum page size

**Problem 3 (5 points):** A system has four processes and five allocable resources. The current allocation and request matrix are as follows:



Is this a safe state? Show a sequence of scheduling (if exists) that will lead to successful completion of the tasks. The request matrix specifies how many of each type of resources are still needed by each process, in addition to what it currently has.

This is a safe state despite the system not having enough resources for the successful completion of Process B the system does not deadlock.

Sequence of scheduling:

Allocate available resources to Processes D and C since they have sufficient resources in the request matrix to proceed.

After processes D and C are complete free up the resources they were holding

Allocate available resources to processes A

After process A is completed, we still need another resource to satisfy B’s request, but we don't have it as a result this is the maximum number of tasks we can complete.

**Problem 4 (5 points):** A system has two processes and three identical resources. Each process needs a maximum of two resources. Is deadlock possible? Explain your answer.

No, a deadlock is not possible. Deadlock is a result of multiple processes unable to proceed because each is waiting for a resource held by another. Of the 2 processes they both need a maximum of 2 resources. If all the resources are used one process is guaranteed to have 2 resources and one process is guaranteed to have one. As a result, the first of the processes cannot deadlock as it has the resources it needs to complete, and the second process cannot lock it out since it needs the first proccess needs no further resources.

**Problem 5 (2+3 points):** Bitmap and free list are two ways to keep track of the free space on the disk. Consider a disk of 2 Gigabytes with 2^20 (i.e., 2 to the power of 20) blocks.

a. What will be size of each block?

2 GB -> 2^31 bytes

Size of each block = Total disk size/ Number of blocks

2^31 bytes / 2 ^20 bytes = **2048 bytes** is the size of each block

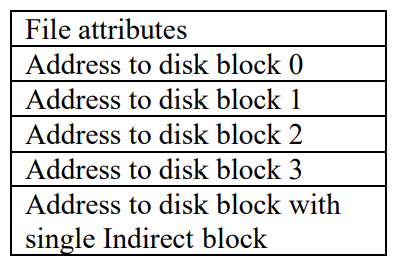
b. If you use bitmap, how much space (in bytes or blocks) will be needed to represent the free space in the disk?

Size of bit map in bytes = Number of bits / 8

Number of bits in bitmap = number of blocks on the disk (to determine free or allocated)

2^20/8 = **2^17 bytes** is the size of the bit map

**Problem 6 (5 points):** Suppose each disk block can hold 256 bytes of data. Assume that it takes 8 bytes to record the address of a disk block. Calculate the maximum file size that can be allowed in a file system assuming the following i-node structure.



Direct block holds = 256 bytes

Total space for direct blocks = 4 \*256 = 1048 bytes

Single indirect = (disk block size / address size) \* disk block size =(256/8) \* 256 = 32 \* 256 = 8192

Total = direct + indirect = **9216 bytes** maximum file size

**Problem 7 (3+2 points):** Let us assume that a file system uses 4-KB disk block size.

a. In your system, if all files were exactly 3 KB, what fraction of the disk space would be wasted?

Maximum one file per block: 1- ¾ (file size/disk block size) = 0.25 or 25% of disk space wasted

b. In your system, assume that the median file size is 12 KB. Will the wastage for this file system be higher or lower compared to scenario (a)? Explain.

In part A there was 1KB of wasted space per block. However, with an average file size of 12KB there are significant odds that the files are likely to fill up multiple blocks completely. As a result, the wastage for this file system will be much lower compared to scenario a.

**Problem 8 (5 points):** How many disk operations are needed to fetch the i-node for the file /usr/ast/courses/os/a.txt? Explain. Assume that the i-node for the root directory is in the memory, but nothing else along the path is in memory. Also assume that all directories for a given i-node fit in one disk block.

5 disk operations are required to fetch the i-node for the file:

1-4. Fetch the i-node for the following directories /usr/ast/courses/os

5. Fetch the i-node for the file a.txt