CSCI 4210 — Operating Systems
CSCI 6140 — Computer Operating Systems
Sample Final Exam Questions (document version 1.0)

Overview

- The final exam will be on **Wednesday, May 10, 2017** from 11:30AM-2:30PM in Sage 3101 and Sage 3303 (please arrive early and sit with empty seats next to you on both sides). Go to Sage 3101 if your last name starts with **A-E**. Go to Sage 3303 if your last name starts with **F-Z**.

- The final exam will be 180 minutes; therefore, if you have 50% additional time, you will have 270 minutes; if you have 100% additional time, you will have 360 minutes.

- The final exam will count as 20% of your final course grade.

- You may bring three double-sided (or six single-sided) 8.5”x11” crib sheets containing anything you would like; crib sheets will not be collected.

- Final exam and final course grades will be made available via Submitty.

- Final exams will **not** be handed back or available per review; however, exams will be graded multiple times to ensure correctness and consistency of grading.

- The final exam will be comprehensive. Therefore, please refer to the sample exam documents for Exam 1 and Exam 2.
Sample Final Exam Questions

1. Contiguous Memory Allocation:

Consider a contiguous memory allocation scheme with dynamic partitioning for a 64MB physical memory with five pre-allocated processes (i.e., A, B, C, D, and E) that just happen to have memory requirements that are evenly divisible by 1MB.

Given new processes F, G, and H) that arrive (almost) simultaneously in the order shown below, show how memory allocation occurs for each of the given placement algorithms.

<table>
<thead>
<tr>
<th>Arrival Order</th>
<th>ProcessID</th>
<th>Memory Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F</td>
<td>2,987,642 bytes</td>
</tr>
<tr>
<td>2</td>
<td>G</td>
<td>4,002,016 bytes</td>
</tr>
<tr>
<td>3</td>
<td>H</td>
<td>6,202,454 bytes</td>
</tr>
</tbody>
</table>

Note that if a process cannot be placed, be sure to state that, then move on to the next process. Do not perform defragmentation.

- For the memory allocation shown below, apply the First-Fit algorithm:

```
AAAAA......B (each character here represents 1MB)
BBBBB......CCCC
CC...DDDDDDDD
DDDD......EEE
```

- For the memory allocation shown below, apply the Best-Fit algorithm:

```
AAAAABBBBBBBBBBB
BBBBBBB.....CCCC
CCCC...DDDDDDDD
DDDDDD.......EEE
```

- For the memory allocation shown below, apply the Next-Fit algorithm, with process D being the last-placed process:

```
AAAAA......B
BBBBBBB.....CCC
CC...DDDDDDDD
DDDD......EEE
```

2. For each of the above algorithms, how much space is unused after the processes are allocated to memory?

3. For each of the above, what kind of fragmentation occurs (i.e., internal or external)?
4. Non-contiguous Memory Allocation:

Consider a non-contiguous memory allocation scheme in which a logical memory address is represented using 32 bits. Of these bits, the high-order 12 bits represent the page number; the remaining bits represent the page offset.

- What is the total logical memory space (i.e., how many bytes are addressed)?
- How many pages are there?
- What is the page size?
- What is the frame size?
- How does logical memory address 23,942,519 (binary 1011011010101010101110111) map to physical memory (i.e., what is the logical page number and page offset)?
- If a process requires 78,901,234 bytes of memory, how many pages will it require?
- How many bytes are unused due to external fragmentation?
- How many bytes are unused due to internal fragmentation?
- Given that the page table is stored entirely in memory and a memory reference takes 100 nanoseconds, how long does a paged memory reference take?
- Adding a translation look-aside buffer (TLB) with a TLB access time of 15 nanoseconds, how long does a paged memory reference take if a TLB hit occurs?
- Given a TLB hit ratio of 84%, what is the effective memory access time (EMAT)?
5. **Virtual Memory:**

Given a page reference string and an n-frame memory, how many page faults occur for the following page replacement algorithms: FIFO; OPT; LRU; LFU.

**Page Reference Stream:**

```
1 8 4 8 4 3 8 3 4 7 1 7 2 3 2 4 2 7 8
```

6. Given a page reference string and a working set delta, identify the working set at the point indicated below.

**Page Reference Stream:**

```
1 8 4 8 4 3 8 3 4 7 1 7 2 3 2 4 2 7 8
```

```
^ |
```

Use a delta of 3, 5, then 8.