Review

Exam

Covers: Lectures 16 to 25 and Reading

Memory Management

Topics you should understand:

1. What is virtual memory and why do we use it?
2. Memory allocation strategies:
   - Contiguous allocation (first-fit and best-fit algorithms)
   - Paging
   - Segmentation
   - Paged segmentation

Memory Management (cont.)

For each strategy, understand these concepts:

- Address translation
- Hardware support required
- Coping with fragmentation
- Ability to grow processes
- Ability to share memory with other processes
- Ability to move processes
- Memory protection
- What needs to happen on a context switch to support memory management

Things you should be able to do:

- Given a request for memory, determine how the request can be satisfied using contiguous allocation.
- Given a virtual address and the necessary tables, determine the corresponding physical address.

Paging

Topics you should understand:

1. What is paging, a page, a page frame?
2. What does the OS store in the page table?
3. What is a TLB? How is one used?
4. What is demand paging?
5. What is a page fault, how does the OS know it needs to take one, and what does the OS do when a page fault occurs?
6. Page replacement algorithms. For each understand how they work, advantages, disadvantages, and hardware requirements.
   (a) FIFO
   (b) MIN
   (c) LRU
   (d) Second chance
   (e) Enhanced second chance
7. How do global and per-process (aka local) allocation differ?
8. What is temporal locality? What is spatial locality? What effect do these have on the performance of paging?

Paging (cont.)

9. What is a working set?
10. What is thrashing and what are strategies to eliminate it?
11. What considerations influence the page size that should be used?

Things you should be able to do:

1. Given a page reference string and a fixed number of page frames, determine how the different replacement algorithms would handle the page faults.

Secondary Storage

1. What is seek time?
2. What is rotational latency?
3. What is transfer time?
4. Disk scheduling algorithms. How do they work, advantages and disadvantages.
   (a) FCFS/FIFO
   (b) SSTF
   (c) SCAN/LOOK
   (d) C-SCAN/C-LOOK
5. What is interleaving?
6. What is disk read-ahead?

Things you should be able to do:

1. Determine the order in which disk requests would be satisfied using a particular algorithm.

I/O Systems

Topics you should understand:

1. How does the OS communicate with I/O devices?
   (a) Polling
   (b) Interrupts
2. What is DMA? How does DMA affect CPU performance?
3. What is a device driver?
4. How do device characteristics affect the interface to a device?
   (a) Character vs. block transfer
   (b) Sequential vs. random access
   (c) Synchronous vs. asynchronous
   (d) Sharable vs. dedicated
   (e) Speed
   (f) Input, output, or both
5. What are I/O buffers used for?
6. What are I/O caches used for? How do they affect reading and writing to I/O devices?

File Systems

Topics you should understand:

1. What is a file, a file type?
2. What types of access are typical for files?
3. What does the OS do on a file open, file close?
4. What is a directory?
5. What is a link?
6. What happens if the directory structure is a graph?
7. How does Unix support multiple users of shared files?
8. Strategies for laying files out on disk. Advantages and disadvantages.
   (a) Contiguous allocation
   (b) Linked
   (c) Indexed

General Skills

- Think about similarities and differences between management of the different levels of the memory and I/O systems.
- You should have a good sense of how the pieces fit together and how changes in one part of the OS might impact another.
- You will **not** be asked to read or write C++ code.
- You will **not** be asked detailed questions about any specific operating system, such as Unix, Windows NT, ...