Fall 2017 Acme-CS241-Review-By-Doing

AKA Preparing for the Final Exam & Beyond CS241...

Some of the questions require research (wikibook; websearch; wikipedia). OK to work together & discuss answers! Be ready to discuss and clear confusions & misconceptions in your last discussion section. NB The final will also include pthread, fork-exec-wait questions and virtual memory address translation. Be awesome.

> 1. C

1. What are the differences between a library call and a system call? Include an example of each.
2. What is the `*` operator in C? What is the `&` operator? Give an example of each.
3. When is `strlen(s) != 1+strlen(s+1)` ?
4. How are C strings represented in memory? What is the wrong with `malloc(strlen(s))` when copying strings?
5. Implement a truncation functionm `void trunc(char*s,size_t max)` to ensure strings are not too long.' Shorter strings are left unchanged. If `s` is `NULL` return `NULL`. e.g. `char s[]="abcdefgh; trunc(s,3); puts(s)` will print "`abc\n`".
6. Complete the following function to create a deep-copy on the heap of the `argv` array. Set the result pointer to point to your array. The only library calls you may use are `malloc` and `memcpy`. You may not use `strdup`.
   `void duplicate(char**argv, char***result);`
7. Write a program that reads a series of lines from `stdin` and prints them to `stdout` using `fgets` or `getline`. Your program should stop if a read error or end of file occurs. The last text line may not have a newline char.

> 2. Memory

1. Explain how a virtual address is converted into a physical address using a multi-level page table. You may use a concrete example e.g. a 64bit machine with 4KB pages.
2. Explain *Knuth*'s and the *Buddy* allocation scheme. Discuss internal & external Fragmentation.
3. What is the difference between the MMU and TLB? What is the purpose of each?
4. Assuming 4KB page tables what is the page number and offset for virtual address 0x12345678 ?
5. What is a page fault? When is it an error? When is it not an error?
6. What is Spatial and Temporal Locality? Swapping? Swap file? Demand Paging?

> 3. Processes and Threads

1. What resources are shared between threads in the same process?
2. Explain the operating system actions required to perform a process context switch
3. Explain the actions required to perform a thread context switch (to a thread in the same process)
4. How can a process be orphaned? What does the process do about it?
5. How do you create a process zombie?
6. Under what conditions will a multi-threaded process exit? (List at least 4)

> 4. Scheduling

1. Define arrival time, pre-emption, turnaround time, waiting time and response time in the context of scheduling algorithms. What is starvation? Which scheduling policies have the possibility of resulting in starvation?
2. Which scheduling algorithm results the smallest average wait time?
3. What scheduling algorithm has the longest average response time? shortest total wait time?
4. Describe Round-Robin scheduling and its performance advantages and disadvantages.
5. Describe the First Come First Serve (FCFS) scheduling algorithm. Explain how it leads to the convoy effect.
6. Describe the Pre-emptive and Non-preemptive SJF scheduling algorithms.
7. How does the length of the time quantum affect Round-Robin scheduling? What is the problem if the quantum is too small? In the limit of large time slices Round Robin is identical to _____?
8. What reasons might cause a scheduler switch a process from the running to the ready state?

> 5. Synchronization and Deadlock

1. Define circular wait, mutual exclusion, hold and wait, and no-preemption. How are these related to deadlock?
2. What problem does the Banker's Algorithm solve?
3. What is the difference between Deadlock Prevention, Deadlock Detection and Deadlock Avoidance?
4. Sketch how to use condition-variable based barrier to ensure your main game loop does not start until the audio and graphic threads have initialized the hardware and are ready.
5. Implement a producer-consumer fixed sized array using condition variables and mutex lock.

6. Create an incorrect solution to the CSP for 2 processes that breaks: i) Mutual exclusion. ii) Bounded wait.
7. Create a reader-writer implementation that suffers from a subtle problem. Explain your subtle bug.

> 6. IPC and signals

1. Write brief code to redirect future standard output to a file.
2. Write a brief code example that uses `dup2` and `fork` to redirect a child process output to a pipe
3. Give an example of kernel generated signal. List 2 calls that can a process can use to generate a `SIGUSR1`.
4. What signals can be caught or ignored?
5. What signals cannot be caught? What is signal disposition?
6. Write code that uses `sigaction` and a signal set to create a `SIGALRM` handler.
7. Why is it unsafe to call `printf`, and `malloc` inside a signal handler?

> 7. Networking

1. Explain the purpose of `socket`, `bind`, `listen` accept functions
2. Write brief (single-threaded) code using `getaddrinfo` to create a UDP IPv4 server. Your server should print the contents of the packet or stream to standard out until an exclamation point "!" is read.
3. Write brief (single-threaded) code using `getaddrinfo` to create a TCP IPv4 server. Your server should print the contents of the packet or stream to standard out until an exclamation point "!" is read.
4. Explain the main differences between using `select` and `epoll`. What are edge- and level-triggered `epoll` modes?
5. Describe the services provided by TCP but not UDP.
6. How does TCP connection establishment work? And how does it affect latency in HTTP1.0 vs HTTP1.1?
7. Wrap a version of `read` in a loop to read up to 16KB into a buffer from a pipe or socket. Handle restarts (`EINTR`), and socket-closed events (return 0).
8. How is Domain Name System (DNS) related to IP and UDP? When does host resolution *not* cause traffic?
9. What is NAT and where and why is it used?

> 8. Files

1. Write code that uses `fseek`, `ftell`, `read` and `write` to copy the second half of the contents of a file to a pipe.
2. Write code that uses `open`, `fstat`, `mmap` to print in reverse the contents of a file to `stderr`.
3. Write brief code to create a symbolic link and hard link to the file `/etc/password`
4. "Creating a symlink in my home directory to the file `/secret.txt` succeeds but creating a hard link fails" Why?
5. Briefly explain permission bits (including sticky and setuid bits) for files and directories.
6. Write brief code to create a function that returns true (1) only if a path is a directory.
7. Write brief code to recursive search user's home directory and sub-directories (use `getenv`) for a file named "`xkcd-functional.png`' If the file is found, print the full path to stdout.
8. The file `'installmeplz'` can't be run (it's owned by root and is not executable). Explain how to use `sudo`, `chown` and `chmod` shell commands, to change the ownership to you and ensure that it is executable.

> 9. File system

Assume 10 direct blocks, a pointer to an indirect block, double-indirect, and triple indirect block, and block size 4KB.

1. A file uses 10 direct blocks, a completely full indirect block and one double-indirect block. The latter has just one entry to a half-full indirect block. How many disk blocks does the file use, including its content, and all indirect, double-indirect blocks, but not the inode itself? A sketch would be useful.
2. How many i-node reads are required to fetch the file access time at `/var/log/dmesg` ? Assume the inode of (`/`) is cached in memory. Would your answer change if the file was created as a symbolic link? Hard link?
3. What information is stored in an i-node? What file system information is not?
4. Using a version of `stat`, write code to determine a file's size and return `-1` if the file does not exist, return `-2` if the file is a directory or `-3` if it is a symbolic link.
5. If an i-node based file uses 10 direct and `n` single-indirect blocks (1<= `n` <=1024), what is the smallest and largest that the file contents can be in bytes? You can leave your answer as an expression.
6. When would `fstat(open(`*path*`,O_RDONLY),&s)` return different information in s than `lstat(`*path*`,&s)`?

> 10. What would you ask?

Create a hard but fair 'spot the lie/mistake' multiple choice or short-answer question.(Ideally, 50% can get it correct)