

600.211: Unix Systems Programming

Midterm 2

Peter H. Fröhlich
phf@cs.jhu.edu

November 22, 2005
Time: 40 Minutes

Start here: Please fill in the following important information using a **permanent pen** before you do **anything** else! Your exam will **not** be graded if you use a pencil or erasable ink on this page.

Name (print): _____

Login (print): _____

Ethics Pledge: With your signature you **certify** the information above and you also **affirm** the following:
“I agree to complete this exam without unauthorized assistance from any person, materials, or device.”

Signature: _____

Date: _____

Instructions: Please read these instructions carefully before you start. **Switch off** your phones, pagers, and other noisy gadgets! You are **not** allowed to have anything but a pen (pencil, eraser) and this exam on your desk. You are **not** allowed to talk to anyone during the exam. If you have a question, please raise your hand **quietly**. You must **remain seated quietly** until all exams have been collected. Remember that you can **not** claim grading errors if you do not use a **permanent** pen for your answers.

Do not open before you are told to do so!

You got _____ out of 40 points.

1 Binary Warmup

(10 points)

For each of the following statements, determine whether it is either **true** or **false**. (1 point each)

1. In C, the instructions `a[i++] = i` and `a[i] = i+1` have the same effect on array `a`.
2. The “execute” bit has meaning for files but not for directories.
3. A session is a group of processes.
4. A “hole” in a file is created by seeking beyond the current end of the file and writing data.
5. Advisory record locking is enforced by the kernel on each and every file-related call.
6. Sockets can be set to “linger,” that is stay active for `listen()` calls even after they are closed.
7. On BSD-based UNIX systems, asynchronous I/O is usually available only for sockets and terminals.
8. Each `pthread_create()` call results in a new kernel-level thread.
9. We cannot change the disposition of the `SIGALRM` signal.
10. The network socket API does not use any of the same signals that other IPC mechanisms use.

2 Tough Choices

(8 points)

For each of the following questions, circle **one** answer out of the choices given. (2 points each)

1. What is so special about the `access()` call as far as **permission checks** are concerned?
 - (a) It uses the effective user and group ID.
 - (b) It uses the real user and group ID.
 - (c) It allows us to temporarily avoid permission checks.
 - (d) All of the above.
 - (e) None of the above.
2. The **record locks** held by process p on file f are **released** when which of the following events occur?
 - (a) Process p exits cleanly.
 - (b) Process p closes a file descriptor for file f .
 - (c) Process p aborts due to a segmentation fault.
 - (d) All of the above.
 - (e) None of the above.
3. The facilities for **memory-mapped I/O** allow us to **avoid** which of the following system calls?
 - (a) The `ioctl()` and `fcntl()` calls.
 - (b) The `read()` and `write()` calls.
 - (c) The `open()` and `close()` calls.
 - (d) The `mmap()` and `munmap()` calls.
 - (e) None of the above.
4. The facilities for **asynchronous I/O** rely on which of the following **IPC** mechanisms?
 - (a) Pipes.
 - (b) Mutexes.
 - (c) Ports.
 - (d) Semaphores.
 - (e) None of the above.

3 Short Answer

(8 points)

For each of the following questions, answer in **one to three** sentences, the shorter the better. (2 points each)

1. Explain the **starving writer** problem in UNIX. What's the issue? What mechanisms are susceptible to it? How is it usually resolved by the operating system?

2. Congratulations, you found a job! While reading the company's C code, you run across the instruction `i[a] = 0;` where `a` is an array and `i` is an integer. The code compiles cleanly and not a single test case fails. **Explain.**

3. What is the difference between a **hard** link created with `link()` and a **symbolic** link created with `symlink()`? Draw a picture if that helps, but be sure to **explain** as many differences as you can think of.

4. Explain the `pthread_join()` call and its relationship to `pthread_detach()`. Be sure to cover **both** calls separately as well as their implications for each other!

4 The Phantom Web Server

(14 points)

You have been coerced by the Petroleum Broadcasting Service (PBS) to develop a high-performance web server that can keep up with their almost daily doubling in page hits (lots of interest in those gas coupons they offer for download). Since most of the funds available for PBS go to “parked” Petroleum executives on their way into Congress and not broadcasting facilities, you’re stuck with one decent (1 GB RAM) server box (bad news) but lots of connectivity (good news). Note that PBS serves a huge variety of content, their 2 TB storage system is about 80% full already.

1. Describe the overall architecture of the web server you would design for PBS. Make sure you address concurrency issues in detail. **(6 points)**

2. Without any further assumptions about the distribution of page hits across the PBS website, how would you perform the transfer of content from the server to clients? Make sure you address I/O issues in detail. **(4 points)**

3. Would it help you to know more about the distribution of page hits across the PBS website? What kind of distribution could be helpful? How would you exploit it? What happens if the distribution changes? **(4 points)**

This page is intentionally **mostly** blank in case you run out of space elsewhere. If you ended up here early, please go over **everything** again and remain seated **quietly**! Make sure that the title page is filled out correctly and in **permanent** pen. Maybe you want to "rewrite" your **answers** in permanent pen as well?