Exam 2

CS 5460/6460, Spring 2013

Name: _____

- You have two hours to complete this closed-book exam.
- Electronic devices are not allowed.
- You are permitted to refer to one sheet of hand-written notes (which may be double-sided) and nothing else except your own brain.
- Your answers (unless they are just a number or something) must be in full sentences.
- Write all answers in the provided space; use the back of the exam if necessary.
- Make sure you have all six pages (including this one).
- If you write legibly and clearly show your work, we may be able to give partial credit for incorrect answers.
- Don't forget to write your name on this exam.
- It is intended that you can answer the questions on this exam without making additional assumptions. However, if you must make additional assumptions in order to provide a reasonable answer, go ahead and do so, *but write down the assumptions as part of your answer.*

1) The UNIX `read` system call takes three arguments: a file descriptor to read from, a pointer to the buffer into which data from the file should be read, and a count of the number of bytes to read from the file. Its prototype in C is:

```
ssize_t read(int fd, void *buf, size_t count);
```

`ssize_t` and `size_t` are just typedefs for integers. You should assume that `fd` is open for reading and that `buf` points to a block of writable memory at least `count` bytes long. If `read` returns a positive value, this indicates the number of bytes that it read from the file into the buffer. It is expected that "short reads" may occur where a call to `read` results in fewer than the number of requested bytes being read; this is not an error. A side effect of `read` is to advance the current position in the file by the number of bytes that were read. Finally, recall that `read` returns 0 upon reaching the end of the file and -1 if an error occurs.

Now write a C function called `reliable_read` whose argument list and return type are the same as `read` above. This function must:

- return -1 if any call to `read` returns -1
- attempt to read `count` bytes into the buffer even if one or more short reads occur; if the end of file is not reached and no error occurs, return `count`
- return the number of bytes that were successfully read if the end of file is reached before `count` bytes could be read from the file

2) Describe two advantages and one disadvantage of extent-based file systems. Label each advantage with a "+" and the disadvantage with a "−".

3) Assume that a file system uses blocks that are 512 bytes long, and that each disk block number is stored in a field 32 bits long.

(a) If you are given no additional information beyond the sentence above, what is the largest file that could possibly be stored in this file system?

(b) If an inode has room for 14 direct blocks, 1 indirect block, and 1 doubly-indirect block, then what is the largest file that could be stored in this file system?

4) This question is about the *clock algorithm* for page replacement in a virtual memory system; recall that it is a 1-bit approximation of LRU using the *referenced bit* attached to each page. For convenience, you may assume that an iterator `P` exists where `P.ref` gives access to the referenced bit of the page that `P` currently points at. `P` initially points to the first page of memory. The `P.next` method advances it to point to the next page, eventually cycling back to the beginning after pointing to each page in turn.

Also, there is an array of similar iterators `LP` which gives access to the list of pages belonging to each process. You can use `LP[cur]` to refer to the list of pages beloning to the current process.

(a) Write down the clock algorithm. It should not use the `LP` iterators.

(b) Write down an extended version of the clock algorithm that grabs a page globally (from any process) if the process that needs a new page is using less than N pages and grabs a page locally (from the requesting process) if it is using at least N pages of memory. The N parameter is determined by a system operator. This code should use both the `P` and `LP` iterators.

5) Name one advantage and one disadvantage of a large scheduling quantum. Label the advantage with a "+" and the disadvantage with a "−".

6) Name one advantage and one disadvantage of a large memory page size. Label the advantage with a "+" and the disadvantage with a "−".