

Homework 2

Tom Kelliher, CS 311

30 points, due Feb. 15

1. This problem is based on Programming Problem 2.27 in the textbook. From your Linux account, download the file `fileCopy.c` from the class web site and compile it:

```
gcc -o fileCopy fileCopy.c
```

Using `strace`, generate a system call trace file:

```
strace -o trace ./fileCopy
```

The trace will be stored in the file `trace`. I would suggest simply using `filecopy.c` as the source file.

Answer the eight questions posed in `fileCopy.c`.

2. To get an idea of how an operating system's system calls interact with user programs and device interrupt handlers, you're going to implement, in pseudo-code, four short system calls and two short interrupt handlers. This small "system" uses a pool of buffers to perform I/O between application programs, an input device, and an output device.

The buffers in the buffer pool are in one of three states at any point in time:

- (a) **INFULL** — input buffer full and available to be returned to the user on request.
- (b) **OUTFULL** — output buffer full and waiting to be output to the I/O device when ready.
- (c) **EMPTY** — empty buffer available for either input or output.

These states can be implemented as separate linked lists, which you can initialize and manipulate using, at the pseudo-code level, the functions you implemented in Project 0.

There are four system calls that are invoked by the user to accomplish I/O:

- (a) `bufptr Gbufin()` — returns a pointer to the beginning of the next full input buffer from **INFULL**. This should block if **INFULL** is empty.
- (b) `Rbufin(bufptr)` — places the buffer pointed to by `bufptr` back into **EMPTY**.
- (c) `bufptr Gbufout` — returns a pointer to the beginning of an empty buffer from **EMPTY** into which the user places data to be output. This should block if **EMPTY** is empty.
- (d) `Rbufout(bufptr)` — places the full output buffer pointed to by `bufptr` onto the end of **OUTFULL**.

In addition, there are two interrupt handlers, one for input and one for output:

- (a) **IHinput** — called when an input operation has been completed.
- (b) **IHoutput** — called when an output operation has been completed.

The diagram below illustrates how a buffer can move between these different states, and the role of the system software described above.

