

CS 2204 Lab 11

your name here (please print):

your student ID number here:

Create a subdirectory called `lab11` under your home directory. Perform any necessary work for this lab assignment in that directory. Ensure that this assignment is performed from within a `bash` shell.

1. (0 points) Revisit the lecture notes of Nov 7 and copy the files `cat4.c` and `cat5.c` to your `lab11` directory. Create a `Makefile` to help compile these files using `gcc` into executables called `cat4` and `cat5`. Run these executables and convince yourself they are working as they ought to.
2. (0 points) Then create a massive file for use in testing `cat4` and `cat5`. Rather than opening up a file and typing a lot of characters using `emacs`, write a shell script which uses a loop to dump a lot of characters into the file. You can decide what you want to put inside the file, but it must be a 'big' file. So, how big is big enough? Read on to see how to judge the size of the file.
3. (6 points) Write another shell script called `timer` that runs the two executables you have from step 1, against the massive file created in step 2, with different sizes of the buffer. `timer` must vary the buffer sizes as 2^i for i beginning at zero. For each buffer size and executable, `timer` use `time` to tabulate the time it takes for the executable (record all of user, system, and real times). Your output should be of the following format:

```
cat4:
buffer size      user      system  real
-----
1                ...      ...      ...
2                ...      ...      ...
4                ...      ...      ...
```

```
cat5:
buffer size      user      system  real
-----
1                ...      ...      ...
2                ...      ...      ...
4                ...      ...      ...
```

To be confident about your `time` estimates, inside `timer`, you could run `time` multiple times (say 3-5) even for the same value of buffer size and executable, and average these values into a single number that you then print in your output (this averaging step is optional). You are free to choose how to structure `timer`, i.e., whether it takes any command line arguments, what they are, any functions inside the script, or the UNIX commands it uses.

4. (4 points) Compare the tabulations across the two executables and make observations about their relative efficiency, as the buffer size varies. Write your observations here in the space below.