Midterm: Dec. 3rd

Goal: Write a “Sort” client-server application

Server Flow:

1) Create an <info> file (e.g. */tmp/sortinfo*) containing:
   - Its own process id.
   - The path to a file (e.g. */tmp/infile*).
2) Wait for a signal (e.g. SIGUSR1)
3) Open the file, whose name was published in the <info> file, (e.g. */tmp/infile*) and read its content, assuming it is a list of paths to other files.
4) **fork()** & **exec()** the **sort** command, passing as arguments the names of files read in the previous step
5) Return the sorted file into a file named as the first item in the input file
   - Hint: use **execv()** and look at “**-o**” option of **sort** to create a temporary file, which will be moved upon termination
6) Remove the input file (e.g. /tmp/infile)

Client Flow:

1. Get names of input files as command line arguments
2. Read the server process id. and input filename from the <info> file (checking for possible errors)
3. Attempt to create the input file, writing the filenames one per line, managing possible race conditions
   - Hint: check **open()** flags
4. Send a signal (e.g *SIGUSR1*) to server

Example

```
$ sserver &

$ cat /tmp/info
123456
/tmp/infile

$ sclient ./p.txt ./q.txt /home/giorgio/r.txt

$ cat /tmp/infile
./p.txt
./q.txt
/home/giorgio/r.txt
```