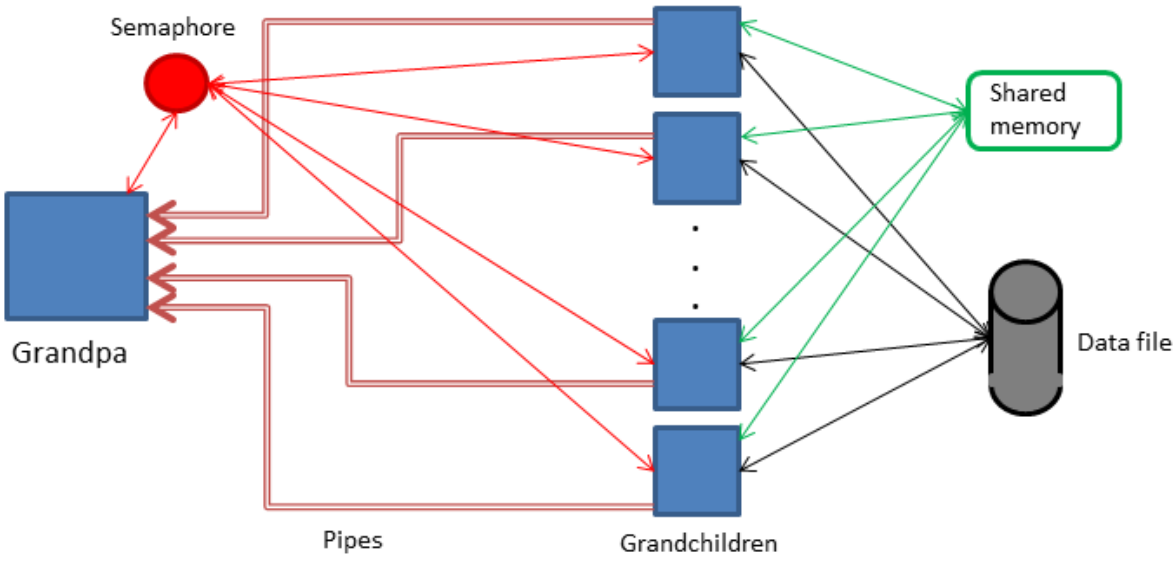
HW-7 CS451 100 points Due: December 4

This is the story of the stash of candy and the grandchildren (and a Grandpa).

Now Grandpa and the 9 grandchildren went for a stroll in Walmart one day, looking for some candy to buy. Lo and behold they found a lot of candy. Their eyes grew huge with desire. But, Grandpa knew that their eyes were much bigger than their tummies, so he set some rules…

Not quite as sophisticated as the five dining philosophers, but it will suffice. We will simulate the feast as follows:




Grandpa will get the data filename as a command-line argument. Grandpa will then create the 9 pipes, 1 semaphore and 1 shared memory (initialized to 0). Grandpa will then make sure the semaphore is free (unlocked). Grandpa will then fork & exe the nine Grandchildren and pass on to them the data filename, the pipe identifiers, semaphore identifier, and the shared memory identifier as command-line arguments.

Note that since Grandpas are not very creative, Grandpa simply named his Grandchildren: p1, p2, … p9.

Note that the data file will just be a file of characters, so you might want to test on your code on something like:

```
Once upon a midnight dreary, fingers cramped and vision bleary,
System manuals piled high and wasted paper on the floor,
Longing for the warmth of bedsheets,
Still I sat there, doing spreadsheets:
Having reached the bottom line,
I took a floppy from the drawer.
Typing with a steady hand, I then invoked the SAVE command
But got instead a reprimand: it read "Abort, Retry, Ignore"
```

Whenever a Grandchild attempts to run, it will first check the semaphore and if it is free (unlocked) it will lock the semaphore and continue; otherwise, it must wait until the semaphore is unlocked. A Grandchild that gets to continue/run will then generate a random number N between 1 and 10 (inclusive) and read N characters from the file, pass its name (p1, etc) followed by those characters to Grandpa, and add the value of N to the value in the shared memory.

Grandpa will extract that data from the pipe, write it to a file (hw7.out), and free the semaphore. Thus, the hw7.out file will have, for example, characters such as:

_p1hel_p4lo

Note the format REQUIRES an underscore in front of the grandchild name.

Once all of the candy (data) have been bought and consumed, the Grandchild who got the last morsel, needs to pass the characters back to Grandpa, acquire the value from the shared memory and pass it back to Grandpa (who will then append it to the file), and then somehow signal the rest of Grandchildren that the candy (data) are/is all gone so they can ALL shut down (take a siesta?). Note that this Grandchild may generate a random number larger than the number of characters left, you will need to watch for EOF to prevent a seg fault. Also, you could encounter the EOF before reading all of the characters, but the number returned to Grandpa will **always** be equal to the actual number of characters read. Finally, Grandpa will append onto the output the file the total number of characters read by all of the Grandchildren (which should also equal the number of characters in the file).

So, on completion, the hw7.out file might look like:

_p1hel_p4lo5

This way Grandpa will have a record of who is the biggest pig of the bunch!

NOTES:
---

1. You really only need one child, named p.c, but you can create 9 children (named pN.c or pN.cpp, where N is 1 through 9) if you want. Or, you can simply spawn the 1 child 9 times.

WARNING:
---

I strongly suggest that you test your code on multiple data sets with multiple runtimes each. If it fails on any of our test sets or runs, will result in, at least, -50% of the grade.

REQUIREMENTS:

---

2. Your program MUST run on the cs server (cscluster.cs.und.edu: 172.16.234.133). We will NOT test it anywhere else.
3. Your full name must appear as a comment at the beginning of your program.
4. Your parent code must be named hw7-yourname.c or hw7-yourname.cpp
5. You must make a tarball of the programs and name that tarball hw7-yourname.tar
6. You must include a makefile to correctly make the parent and children with whatever executable names you desire/require.
7. Failure to follow the above will result in, at least, -50% of the grade.