Santa Clara University

Department of Computer Engineering

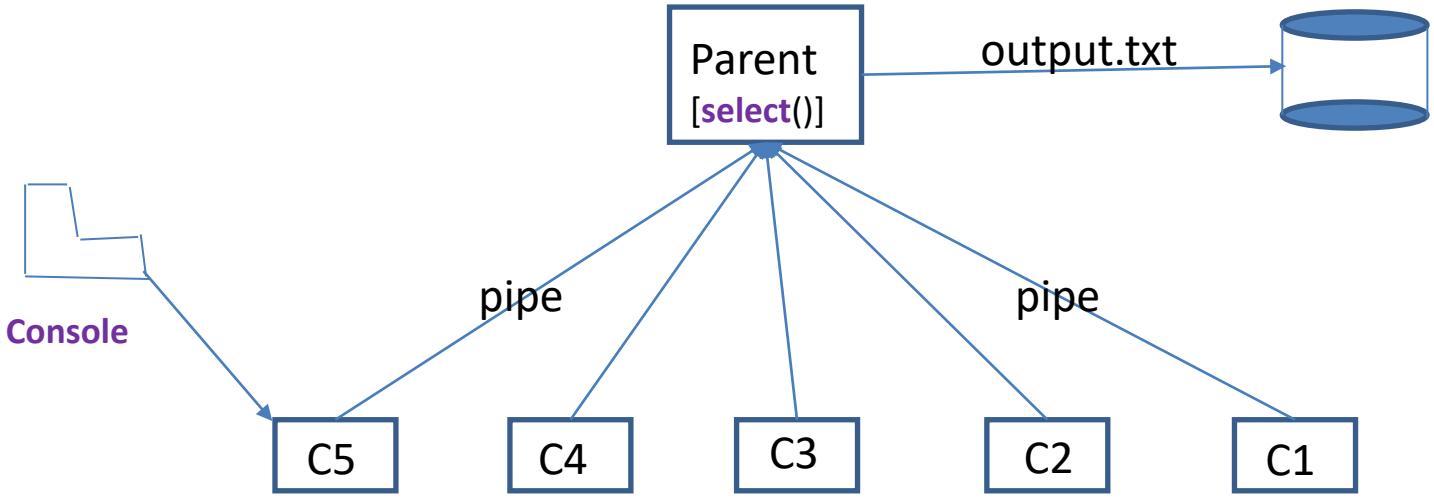
Advanced Operating Systems (COEN 383)

Project-6 Preview (6 pts)

Instructor: Ahmed Ezzat

UNIX/Linux I/O

The purpose of this assignment is to practice making UNIX I/O system calls in C. In a multiplexed manner, your main process (parent) will read from multiple pipes, from the standard input (the terminal), and write to the file system.



Homework-6 Architecture

A parent process will create/fork() five children processes with a dedicated pipe between the parent and each of the 5 children. All five children each has life time equals 30 seconds. Four children behave one way and the fifth behave differently. (1) Each of the four children sleeps for a random amount of time (0, 1, or 2 seconds) and when the process wakeup, it generate a time stamped message to the nearest 1000th of a second (gettimeofday(&tv, NULL)), sends it to the parent process through the dedicated pipe, then the child process go back to sleep. When

the 30 seconds expires, the child close the pipe() and exits. A message should have the following format:

0:00.123: Child 1 message 1 // 0.123 mseconds
0:02.456: Child 1 message // 2.456 mseconds

#include <sys/time.h>

int gettimeofday(struct timeval *_tv_, struct timezone *_tz_);

If either tv or tz is NULL, the corresponding structure is not returned.

```
struct timeval {
    time_t      tv_sec;     /* seconds */
    suseconds_t tv_usec;    /* microseconds */
};

int     sec  = (int) (tv.tv_sec);
double  msec = ((double)(tv.tv_usec)/1000);
```

(2) The 5th child process interact with the login terminal: the child process prompts (stdout) the terminal to read the next input string from the stdin. Child reads the input string from stdin typed by the user. Write that string with a timestamp to the parent through the dedicated pipe, and then immediately prompt for the next message. Terminate the process after 30 seconds.

s:msec:Child X: Nth text msg from the terminal
0:00.123: Child 5: 1st text msg from the terminal
0:01.450: Child 5: 2nd text msg from the terminal

The parent uses the select() system call to read any available messages from its 5 children, append current time stamp to each message before writing to the output file (output.txt). Terminate the parent process after all children have exited/terminated

P.S. when the child (sending process) closes the write fd of the pipe on exit (say Pipe-1), the receiving process (parent) using select() will get indication that it has message on the appropriate end of that pipe (Pipe-1) and when it reads it returns 0, indicating EOF. The parent knows now that the child exited.

What to turn in

Email a zip file to your grader, Name the zip file after your team Group#, for example, **Group-3.zip**. The file should contain:

- Your C source files
- Your **output.txt** file
- A brief report describing any issues you encountered.