Lab 2: xv6 and process

Instructor: Yaoqing Liu

TA: Garegin Grigoryan

Xv6 Review

- ssh to odin and change directory to Lab2

$ssh YourName@odin.cslabs.clarkson.edu

$cd ~/cs444-s18/Lab2

- Download lab2.tar.gz file to your work directory Lab2

$wget http://people.clarkson.edu/~liu/CS444/Spring18/Lab2.tar.gz

- Unzip it

$tar –xzvf Lab2.tar.gz

Compile xv6 code

- Go to xv6 directory and compile xv6

$make

- Run xv6 operating system

$make qemu-nox

Test a user program

$ls

$cat README

Quit xv6

- Two steps to quix xv6
- Ctrl + A, C
- $quit

API introduction

- Fork()
- Wait()
- Exec()
- Exit()
- File operations
  - 0 – standard input
  - 1 – standard output
  - 2 – standard error

  - Close(0)
  - open(“test.txt”, O_RDONLY) [cat 0 === cat test.txt]
  - Arg[0]=“cat”
  - Arg[1]= 0
  - Exec (“cat”, arg)

Demo

- Fork, wait, exec, file operation

Your Tasks

- Write a xv6 user program that does the following tasks
  - A program with three processes: the parent process itself, one child process and one grand child process
  - The grand child process creates and opens a file called test.txt and write “Hello, Your Name” to it
  - The child process check if the file exists using “ls” command
  - The parent process outputs the content of “test.txt” using command “cat”
  - The child process runs after the grand child process, the parent process runs after the child process
  - All processes should exit properly

How to create your first user program

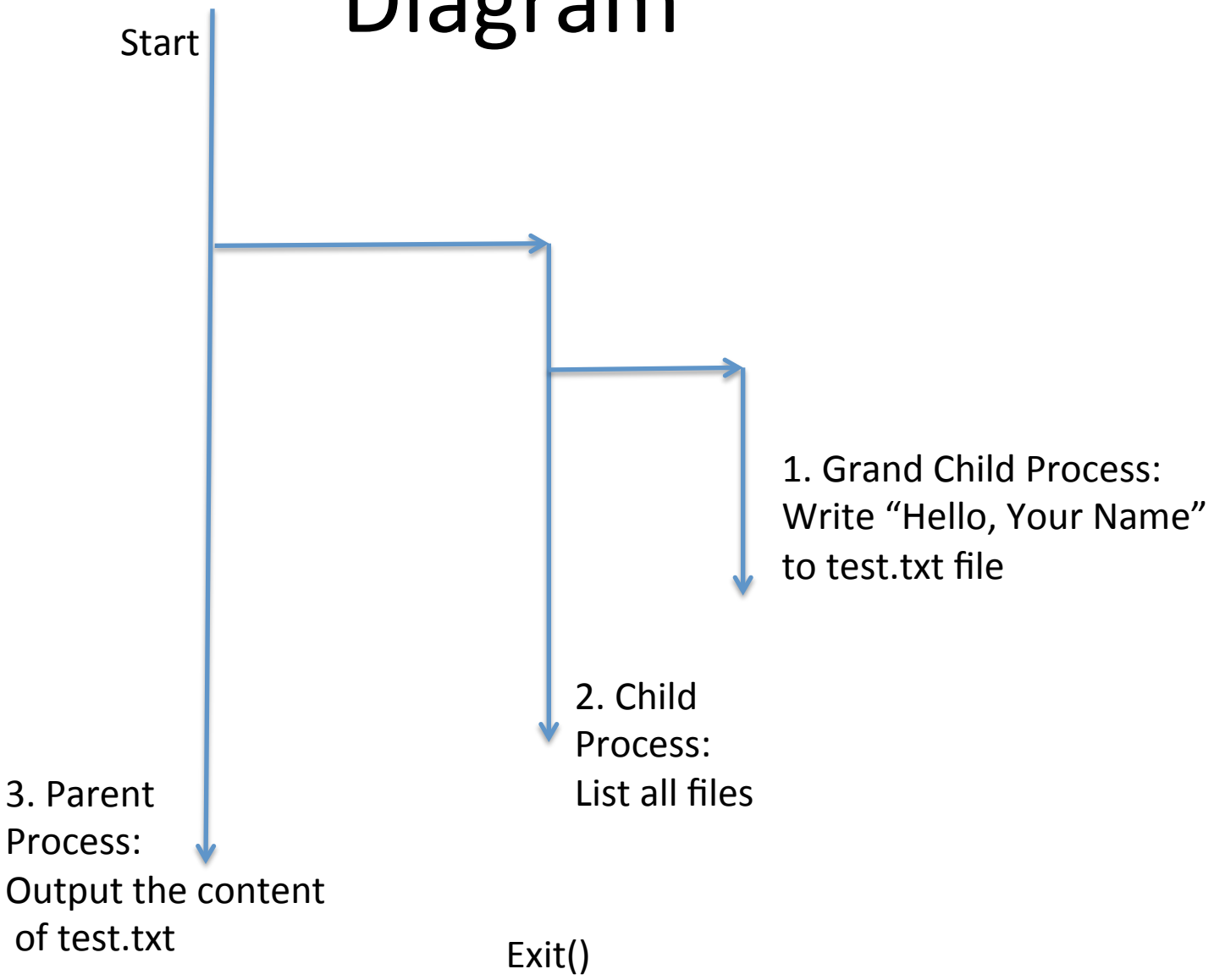
- Change directory to ***userspace*** and create a file with yourname.c
  - You can copy 1_fork.c to yourname.c for test

  $cp 1_fork.c yourname.c
- Open Makefile: extend **UPROGS**=…with _yourname
- Change directory to the parent folder and compile your code, and run it
- $cd ..
- $make
- $make qemu-nox

After this was successfully done, test your own program

- $yourname

Diagram


Start
1. Grand Child Process:
Write “Hello, Your Name”
to test.txt file
2. Child
Process:
List all files
3. Parent
Process:
Output the content
of test.txt
Exit()

Output

```
init: starting sh
$ lab2
Grand Child's branch, start writing to file
Child's branch, list all files
.              1 1 512
..             1 1 512
README         2 2 2487
1_fork         2 3 13336
4_exec         2 4 13400
cat            2 5 14240
echo           2 6 13284
fileopt        2 7 13332
forktest       2 8 8216
grep           2 9 16032
init           2 10 13832
kill           2 11 13360
lab2           2 12 14044
ln             2 13 13192
ls             2 14 15460
mkdir          2 15 13380
pwd            2 16 13316
rm             2 17 13360
sh             2 18 23776
stressfs       2 19 14036
usertests      2 20 56992
wc             2 21 14828
zombie         2 22 13024
console        3 23 0
test.txt       2 24 12
Parent's branch, print content written by grand child
Hello World
$
```

Submission

- Take screenshots for your program
  - Source code: src.png
  - Compiling process: compile.png
  - Output results: results.png
- Combine them together in a pdf file and submit it to moodle

$convert src.png compile.png results.png YourName.pdf

- Leave your code in the Lab2 directory, we will check it

Remove all your local files when you're done

Both moodle submission and coding are due on Jan 29 (Monday) 11:55pm

Thank You!