# CS302
# Operating System
# Lab 2

## Process, Pipe and Signals

March 14th , 2018

Dongping Zhang

cadongllas@gmail.com

https://github.com/cadongllas/CS302_Lab2

# Process

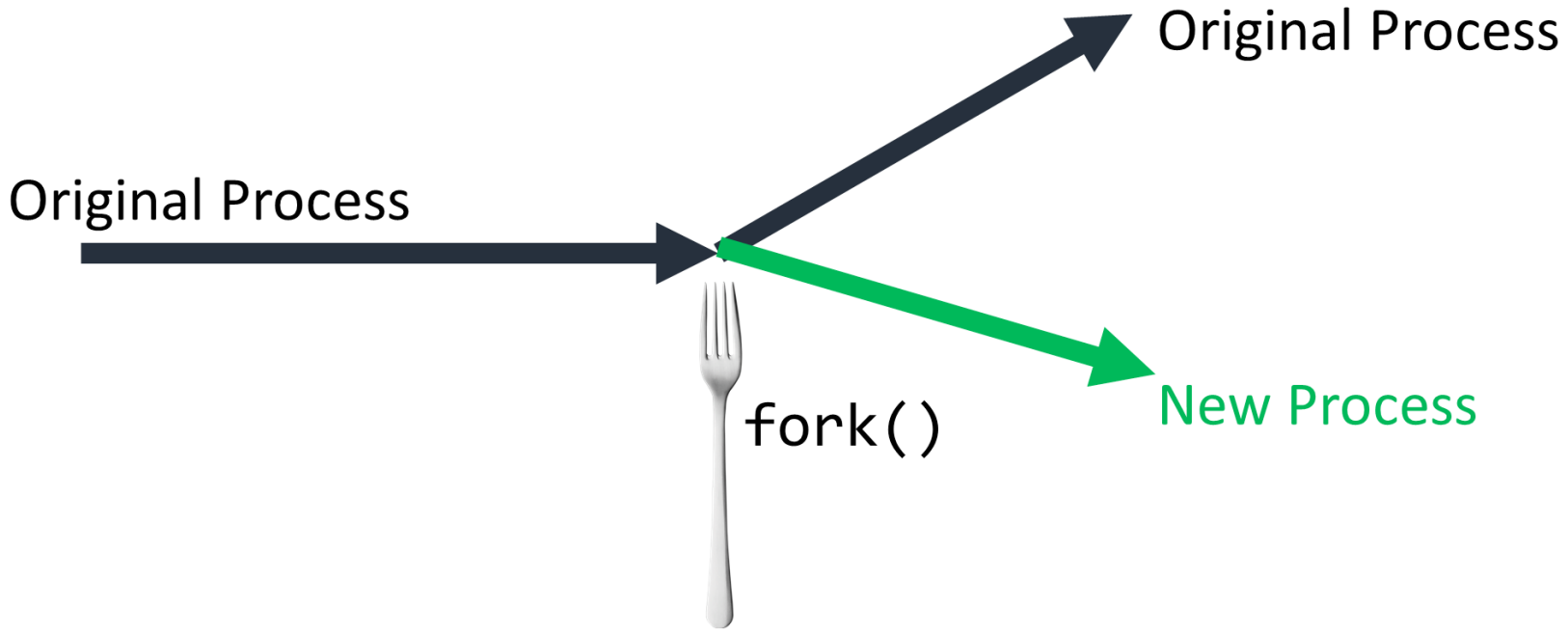
- Process Identification
- Create a process
- Execute a program

# Process Identification

- Checking System Process:
  - top
  - ps
  - ps aux ( a -Shows processes from all users, u -Displays the owner of the process, x -Also shows the processes other than the terminal)
- We can identify one process uniquely by its **Process ID** or **PID**
  - getpid()
  - getpid.c

# Process Creation

- Use system call **fork** to create a process
- after fork, the original process splits into two



# Process Creation

- fork system call
- fork1.c

```
Before Fork, My PID: [8265]
After Fork, My PID: [8265]
After Fork, My PID: [8266]
```

- Distinguish Parent and Child Process
- fork2.c

```
Before fork():PID[2072]
The value of res is 2073
I am parent! PID: [2072]
Program Terminated!
The value of res is 0
I am child! PID: [2073]
Program Terminated!
```

# Process Creation

- Exercises : How many A, B and C will be printed? (fork_ex.c)
  a) A: 1 time, B: 2 times, C: 2 times
  b) A: 1 time, B: 2 times, C: 4 times
  c) A: 1 time, B: 1 times, C: 2 times
  d) A: 1 time, B: 2 times, C: 3 times

```c
#include <stdio.h>
#include <unistd.h>
int main(int argc,char *argv[]){
    printf("A\n");
    fork();
    printf("B\n");
    fork();
    printf("C\n");
    return 0;
}
```

# Process Execution

- execl.c
  - ➢ execl() is one of the family members of exec() for calling external programs.
  - ➢the code of the process is *changed* to the target program and it **never** returns to the original code.
- exec() has several family members.

# Pipes And Signals

- Signals are a kind of interrupt to the running process.

- Pipe is a communication mechanism for sending information between processes.

# Signals

- Send singals using kill
  - **kill** can send all kinds of signals.
  - int kill(pid_t pid, int sig);
  - kill.c
- Custom signal handling
  - sigign.c
  - custom.c

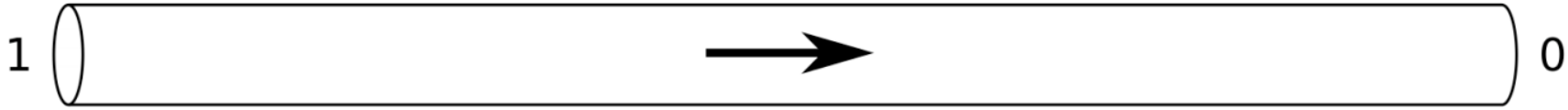
# Pipe

- pipe overview
  - Read the pipe manual pages in section 7
  - Type "man 7 pipe"  (do not need to read the part related to FIFO)

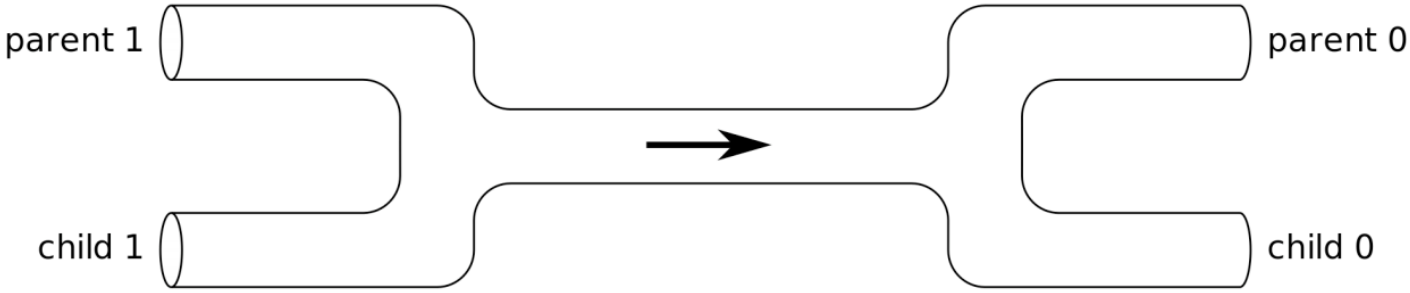
- System call pipe ()
  - pipe is created by system call **pipe()**
  - Type "man 2 pipe" to see detail

# Learn to use pipe

- Pipe Creation
  - Pipe is unidirectional
  - pipe_creation.c

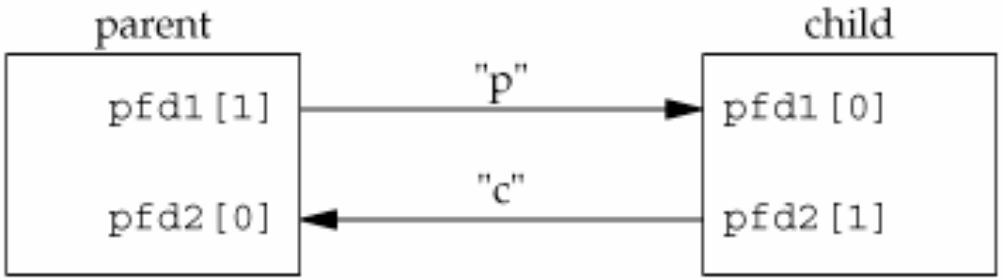



- pipe with fork ()
  - pipe_withfork.c
    - **Always be sure to close the end of pipe you aren't concerned with**
  - pipe_withfork2.c
    - child forget to close write end. What will happen?



# Exercise 1 : Use pipe to synchronize processes

- pipe-ex1.c
- parent and child processes need communicate with each other according to the following rules:
  - ➢ Parent sends a message (a character "p") to child.
  - ➢ Child sends acknowledgment message (a character "c") to parent on receiving "p", otherwise, child waits and does nothing.
  - ➢ When parent receives "c", it sends the next "p" to child, otherwise, parent waits and does nothing.




```
Parent send message to child!
Child receive message from parent!
Child send message to parent!
Parent receive message from child!
Child receive message from parent!
Parent send message to child!
Child send message to parent!
Parent receive message from child!
Parent send message to child!
Child receive message from parent!
Child send message to parent!
Parent receive message from child!
Child receive message from parent!
Parent send message to child!
Child send message to parent!
Parent receive message from child!
Child receive message from parent!
Parent send message to child!
Child send message to parent!
Parent receive message from child!
Alarm clock
```

# Connecting two programs with pipe

- **pipe_lsless.c** implements shell command "ls | less" in C.
- Execrise-2
  - Modify **pipe_lsless.c**, make it accept two inputs from command line and pipe them. Your program should run like this**./pipe4 [command1] [command2]**. You can assume that **command1** and **command2** do not have parameters.
  - when you run your program as**./pipe4 ls less**, the result should be the same with running **ls | less** in shell.

# Report for Lab 2

- Please download the report template and report code and complete the report
- Name your report as **OS_Lab2_XXX_YYYYYYYY.doc**
- Replace XXX with your name and replace YYYYYYYY with your student id
- Suche as **OS_Lab2_张三_12345678.doc**
- Check the **blackboard** for deadline