Pipes and FIFOs
(보충 자료)

Dong-kun Shin
Embedded Software Laboratory
Sungkyunkwan University
http://nyx.skku.ac.kr

Pipe의 중요한 특성

- **마지막 writer가 close 할 때 EOF (End-of-File)가 발생**
  - 즉, pipe의 file descriptor는 필요 없을 때 close 하는 것이 매우 중요하다!!
  - fork() 사용 시 file descriptor table이 복제되기 때문에 close()를 유념해서 사용해야 한다.
    - 부모와 자식 프로세스 모두에서 pipe의 ~~읽기~~**쓰기** 전용 file descriptor를 모두 close 해야만, EOF가 발생하고, ~~writer~~**reader**에게 내용이 전달 된다.
  - dup2()의 작동 원리를 제대로 이해하는 것이 필요하다.
    ex) dup2(fd[0], 0); close(fd[0]);
    - 원래 stdin으로 받는 입력을 fd[0]으로 부터 받게 되고, fd[0]이라는 file descriptor만 닫게 된다. **(fd[0]이 가리키는 file을 닫는 게 아님!)**

Exercise

- **Make C programs run the following tasks:**

$ echo "124 * (42 + 3) % 17" | bc

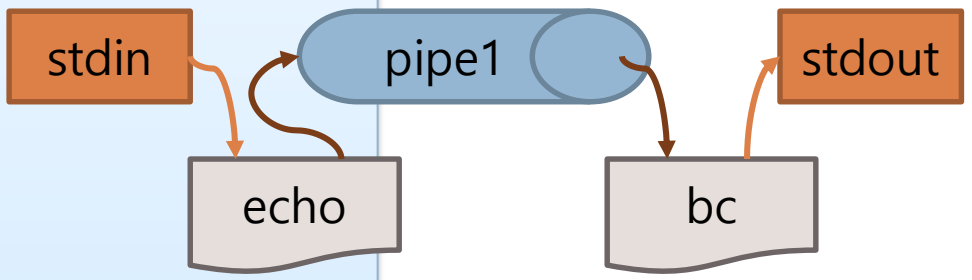
- main -> pipe -> fork

  - dup2 -> exec family ➔ echo
  - dup2 -> exec family ➔ bc

$ cat < /proc/meminfo | grep -i active | tail -n4 > memory.txt

Exercise 1

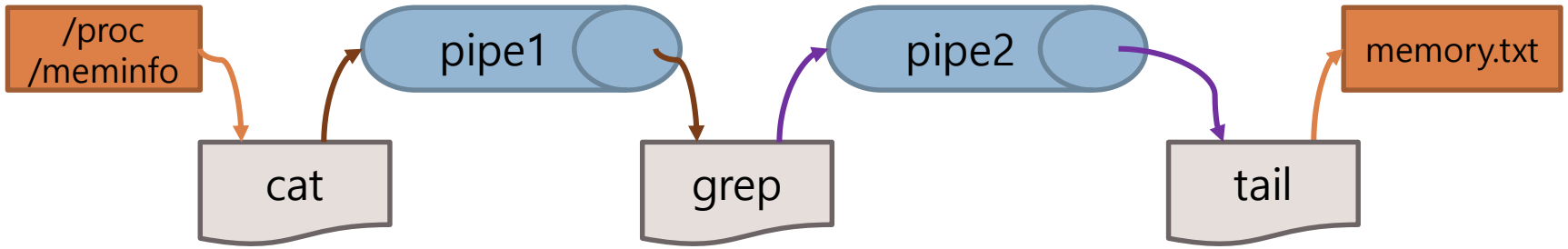

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

#define READ 0
#define WRITE 1

int main(void)
{
  int pipe1[2];
  pid_t pid;
  int state;

  pipe(pipe1);
  if ((pid=fork())==0) {
    dup2(pipe1[WRITE], 1);

    close(pipe1[READ]);
    close(pipe1[WRITE]);

    execl("/bin/echo", "echo", "124*(42+3)%17", NULL);
  }

  close(pipe1[WRITE]);
  if ((pid=fork())==0) {
    dup2(pipe1[READ], 0);

    close(pipe1[READ]);

    execl("/usr/bin/bc", "bc", NULL);
  }

  close(pipe1[READ]);

  waitpid(pid, &state, 0);

  return 0;
}
```



Exercise 2 (1)


```c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <fcntl.h>

#define READ 0
#define WRITE 1

int main(void)
{
  int pipe1[2], pipe2[2];
  pid_t pid;
  int state;

  pipe(pipe1);
  if ((pid=fork())==0) {
    int fd = open("/proc/meminfo", O_RDONLY);

    dup2(fd, 0);
    dup2(pipe1[WRITE], 1);

    close(fd);
    close(pipe1[READ]);
    close(pipe1[WRITE]);

    execl("/bin/cat", "cat", NULL);
  }

  close(pipe1[WRITE]);
  pipe(pipe2);
  if ((pid=fork())==0) {
    dup2(pipe1[READ], 0);
    dup2(pipe2[WRITE], 1);

    close(pipe1[READ]);
    close(pipe2[READ]);
    close(pipe2[WRITE]);

    execl("/bin/grep", "grep", "-i", "active", NULL);
  }

  close(pipe1[READ]);
  close(pipe2[WRITE]);

  if ((pid=fork())==0) {
    int fd = open("memory.txt", O_WRONLY | O_CREAT | O_TRUNC, 0644);

    dup2(pipe2[READ], 0);
    dup2(fd, 1);

    close(pipe2[READ]);
    close(fd);

    execl("/usr/bin/tail", "tail", "-n4", NULL);
  }

  close(pipe2[READ]);

  waitpid(pid, &state, 0);

  return 0;
}
```

Exercise 2 (2)



Good Luck!!

- 질문은 lhy920806@gmail.com으로 보내주세요.