

Pipes

Hyo-Bong Son (proshb@csl.skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
<http://csl.skku.edu>



Pipes

- **The oldest form of UNIX System IPC**
 - Half duplex(data flows in only one direction)
 - Only between processes that have a common ancestor
- **Pipe function**

```
#include <unistd.h>

int pipe (int filedes[2]);
```

- File descriptors are returned through the *filedes* arg.
- *filedes*[0] for reading, *filedes*[1] for writing
- The output of *filedes*[1] is the input for *filedes*[0]

Pipes

■ Example

```
int main(void)
{
    int n, fd[2];
    pid_t pid;
    char line[MAXLINE];

    if(pipe(fd) < 0) perror("pipe error");

    if((pid = fork()) < 0) perror("fork error");
    else if (pid > 0){
        close(fd[0]);
        write(fd[1], "hello world\n", 12);
    } else {
        close(fd[1]);
        n = read(fd[0], line, MAXLINE);
        write(1, line, n);
    }
    exit(0);
}
```

Hoinkies

- **Right hoinky '>' & Left hoinky '<'**

- Redirect input & output

- **Examples**

```
$ ls > output.txt
```

- Redirects std_out to the file "output.txt"

```
$ cat < test.c (same as $cat test.c)
```

- Redirects the file "test.c" to input of cat

Final Mini-Shell

- **Make your mini-shell supports ordinary '>'**
 - `ms::/home/user> ./helloworld > output.txt`
- **Make your mini-shell can redirect a process's output(std_out) to the input(std_in) of another process.**
 - `ms::/home/user> ./helloworld => my_cat`