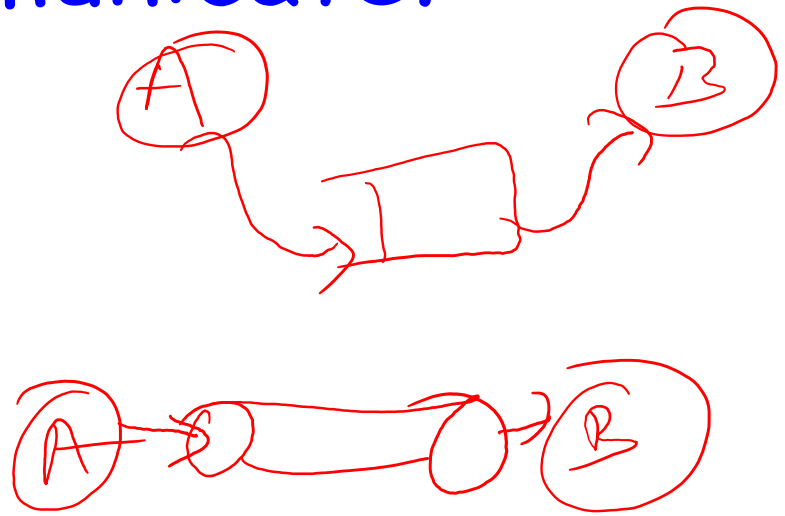
Inter Process Communication
Pipe

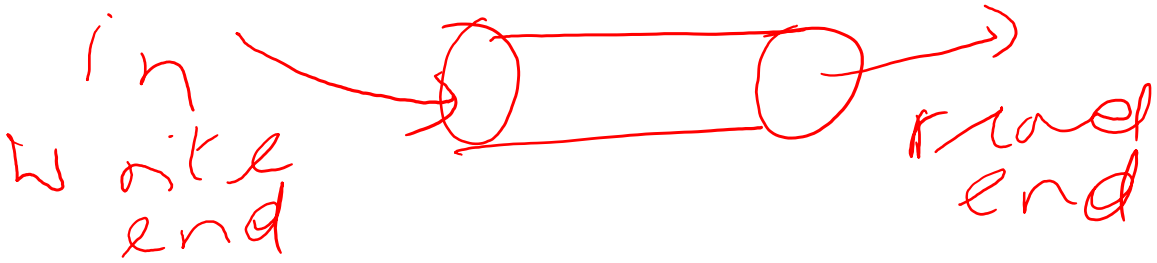
Sridhar Alagar

How can two processes communicate?



- Using files
  - One process can write to a file
  - Another process can read from a file
- Problems:
  - Read can happen before writing
  - How does the reader process know that writing is completed?
  - Hard to synchronize writes and reads

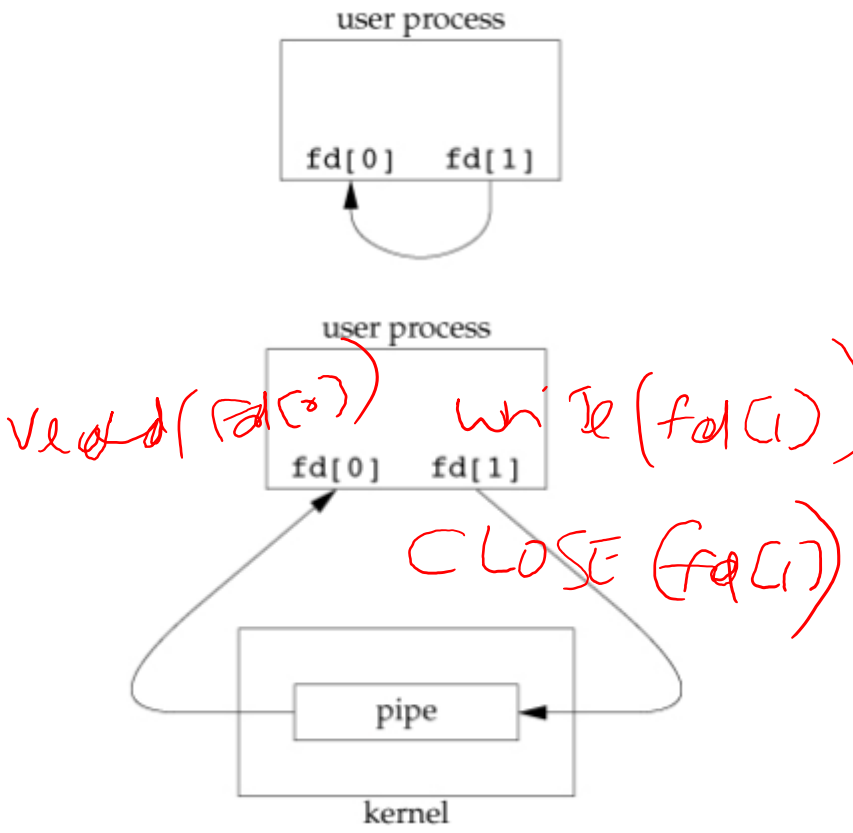
Pipe




```
PIPE(2)


NAME
       pipe, pipe2 - create pipe

SYNOPSIS
       #include <unistd.h>

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

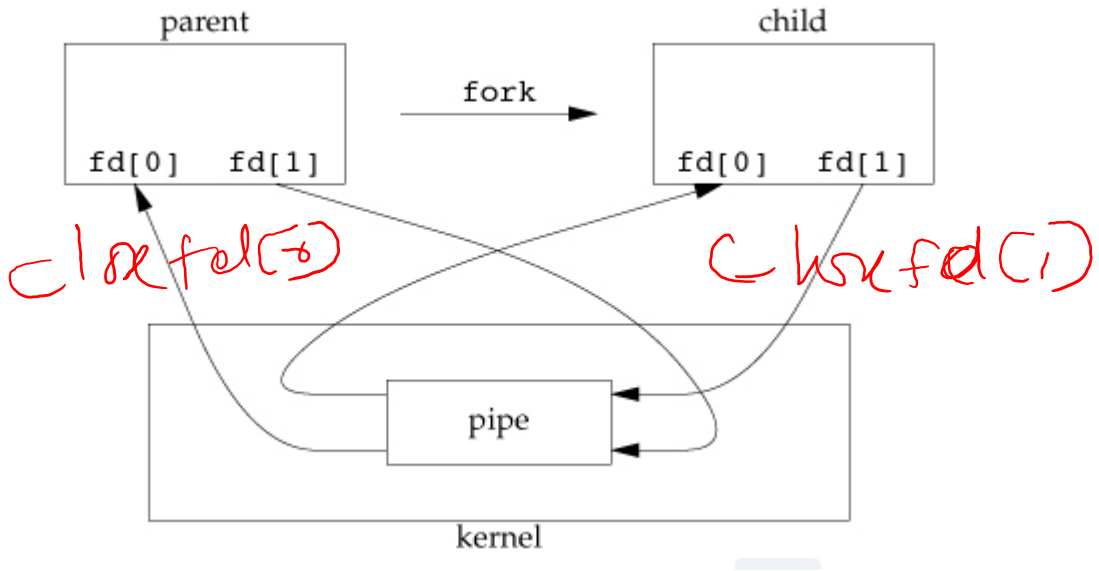
- Creates a unidirectional data channel for IPC
- Two file descriptors are returned
  - pipefd[0] used for reading (use read())
  - pipefd[1] used for writing (use write())

- Data written to the write end of pipe is buffered till the data is read
- Attempt to read from empty pipe is blocked till data is written
- If all descriptors referring to write end of pipe is closed, read() will return 0 (EOF)



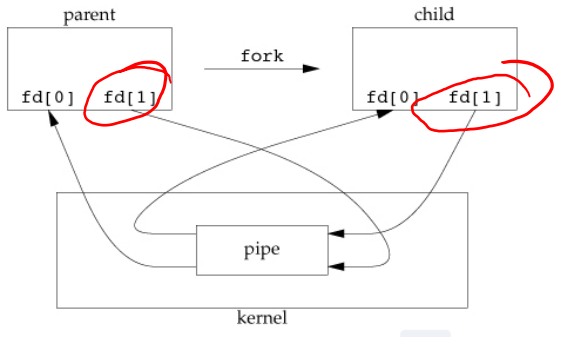
P2C communication using pipe

- pipe(fd) followed by fork()

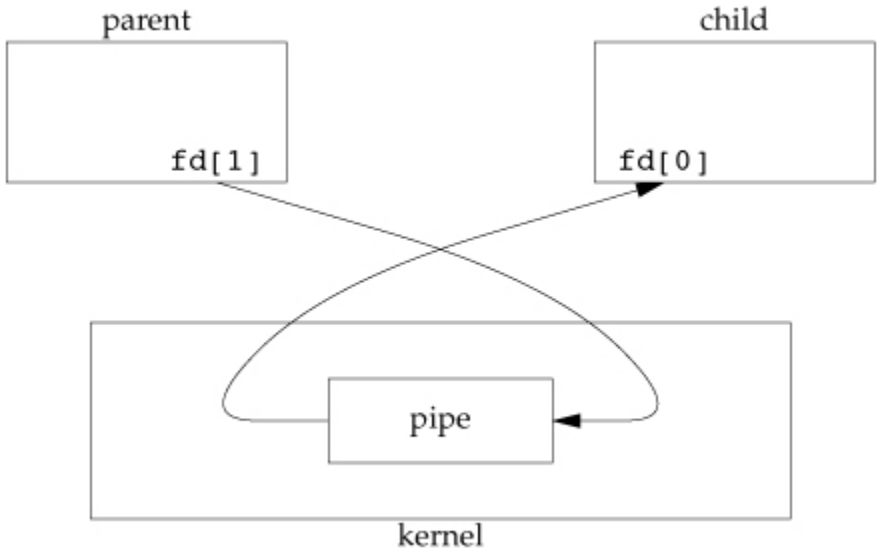


P2C communication using pipe

- pipe(fd) followed by fork()

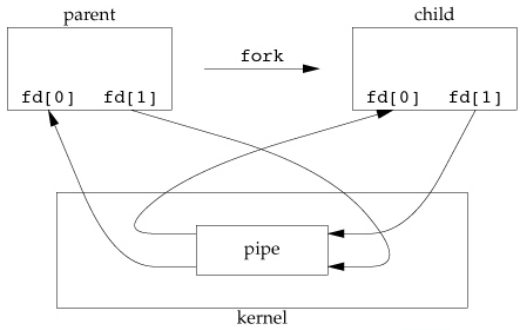


- Establish one way channel from parent to child
- close(fd[0]) in parent - no reading
- close(fd[1]) in child - no writing

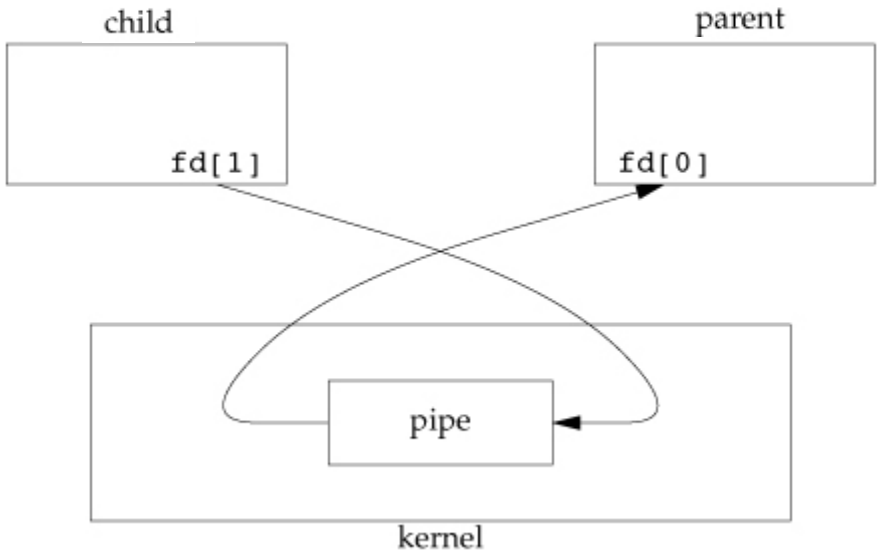


C2P communication using pipe

- pipe(fd) followed by fork()



- Establish one way channel from child to parent
- close(fd[0]) in child - no reading
- close(fd[1]) in parent - no writing



How to implement this piped command?

> cat filename | wc

output of cat is redirected to write end of pipe

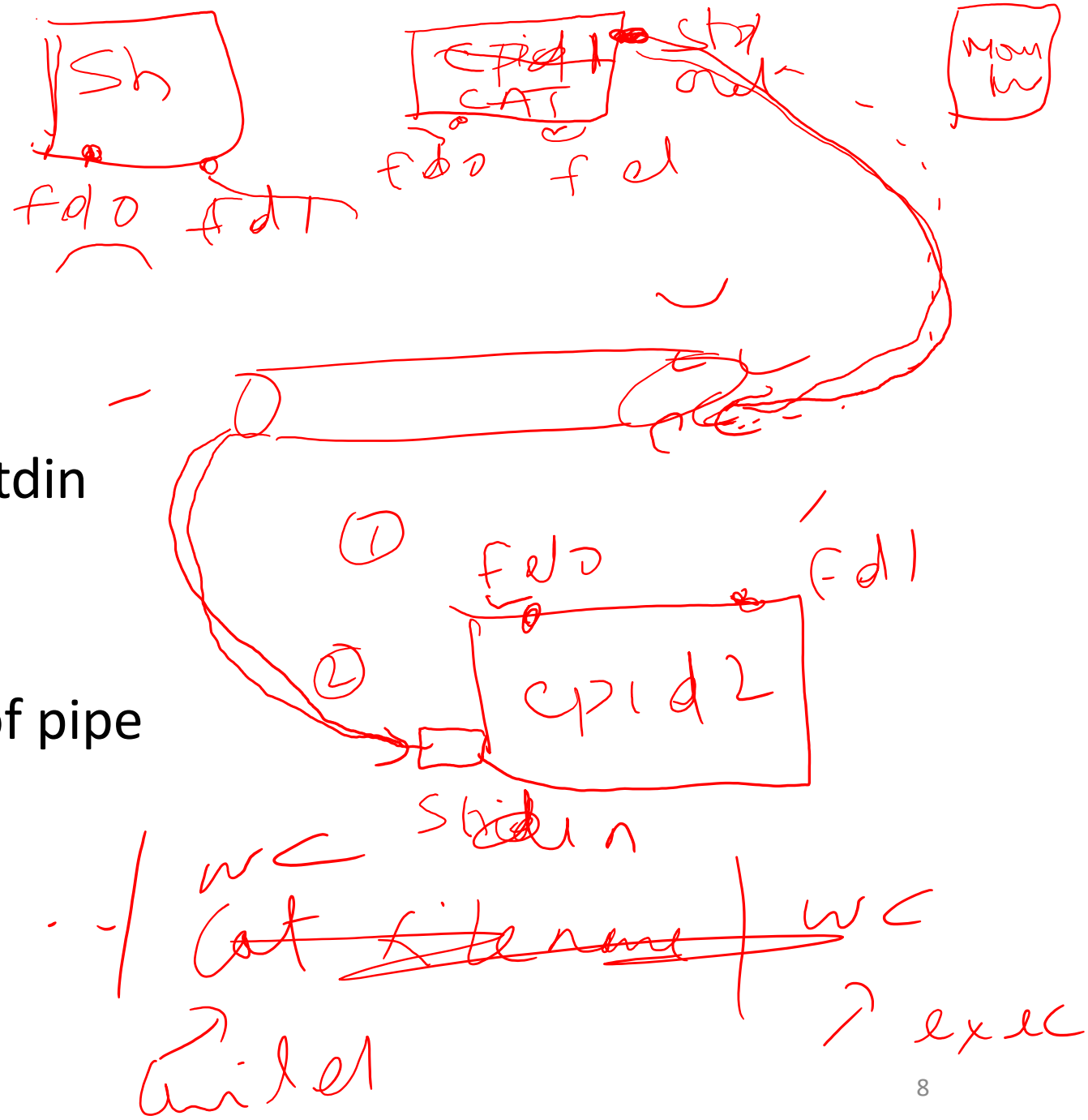
1. fork
2. output redirection
3. execvp cat

redirect read end of pipe to stdin

1. fork
2. input redirection
3. execvp wc

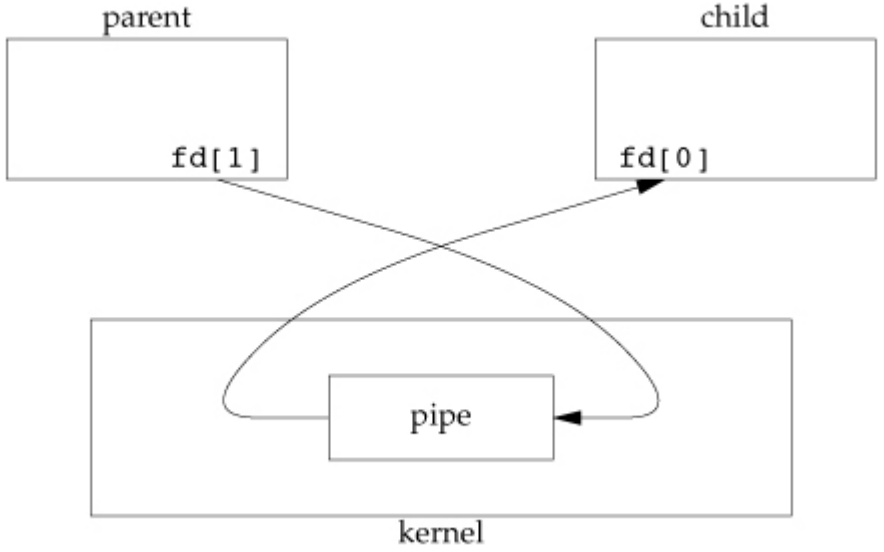
Piped commands

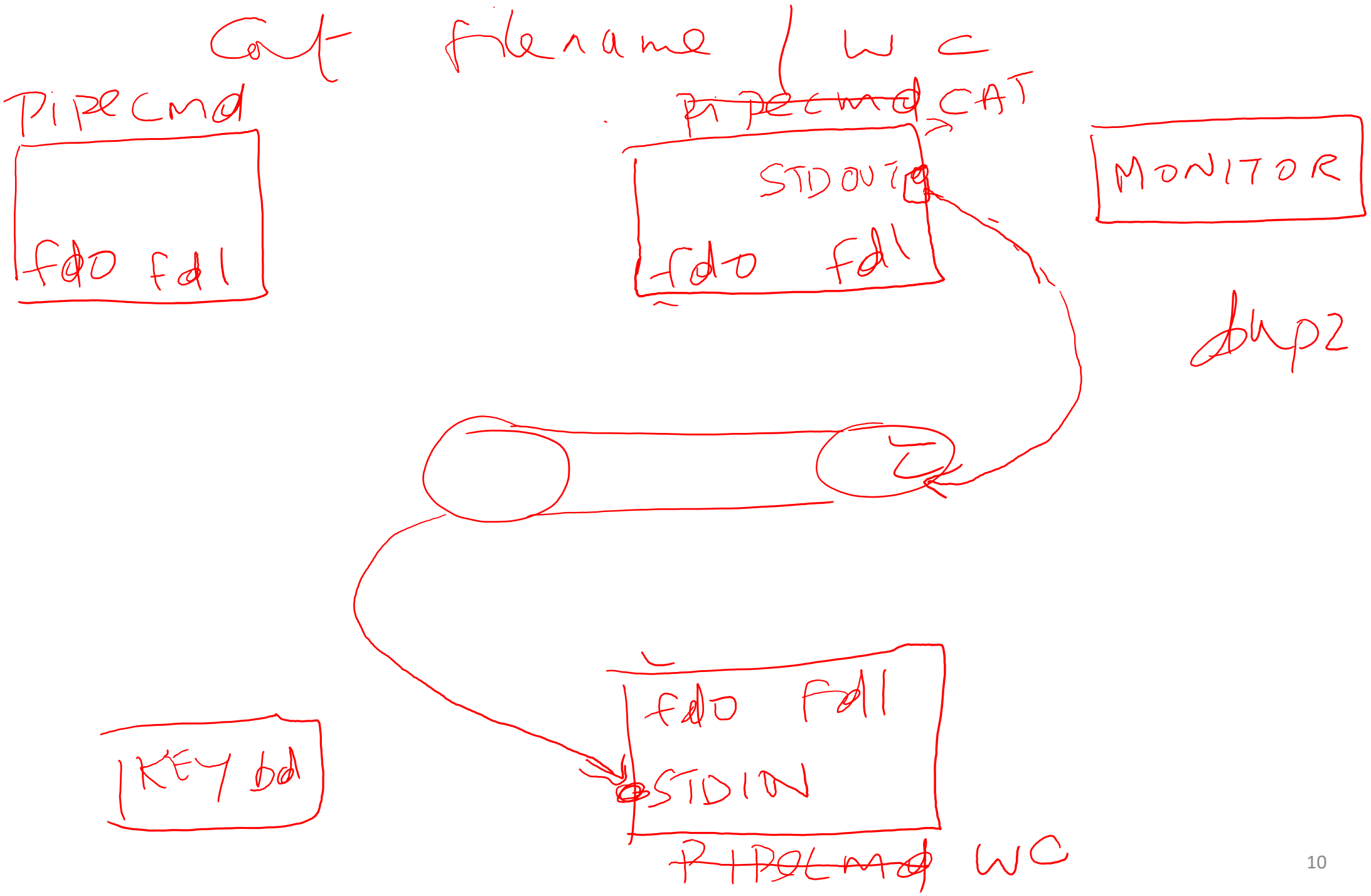
- Use dup2(int oldfd, int newfd)
- To redirect read end of pipe to stdin
  - dup2(fd[0], STDIN__FILENO)
- To redirect stdout to write end of pipe
  - dup2(fd[1], STDOUT__FILENO)

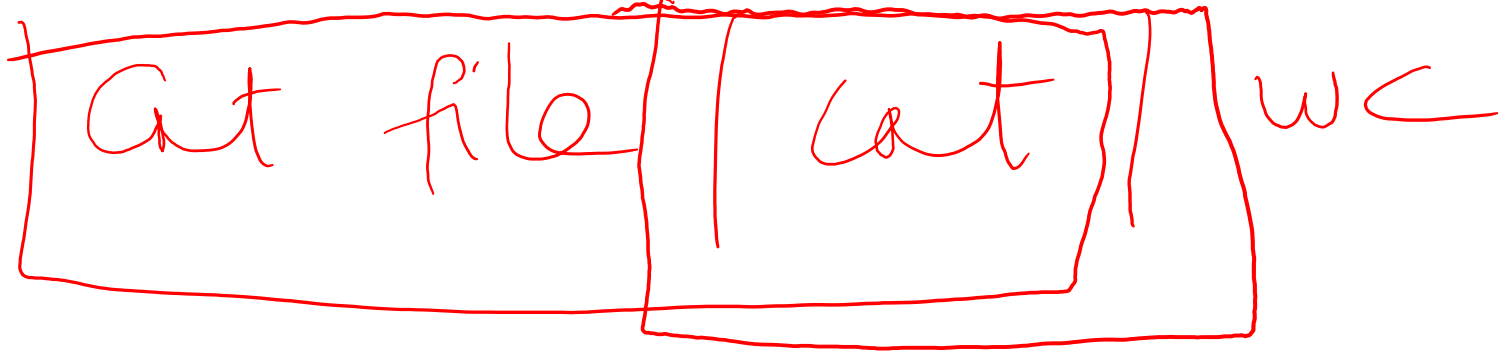


Piped commands

- Use dup2(int oldfd, int newfd)
- To redirect read end of pipe to stdin
  - dup2(fd[0], STDIN__FILENO)
- To redirect stdout to write end of pipe
  - dup2(fd[1], STDOUT__FILENO)




cat filename | wc
Pipecmd
fd0 fd1
Pipecmd CAT
STDOUT
fd0 fd1
MONITOR
dup2
KEYBD
fd0 fd1
STDIN
PIPECMD WC



cat file | cat | cat | cat | cat | . . . | wc

Parent child synchronization