

# Interprocess Communication

CS351 : Saelee

# UNIX IPC

conflicting standards

# System V vs. Posix

**many** mechanisms

1. Signals
2. Process tracing
3. Unnamed & Named pipes
4. Semaphores & File locks
5. Message queues
6. Shared memory
7. Sockets

common issues

unidirectional vs. bidirectional

creating links/endpoints

naming

multiple senders/recipients

speed/capacity

buffering

data synchronization

# Unnamed Pipes

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

`pfid[0]` for reading  
`pfid[1]` for writing

kernel buffer

PIPE\_BUF = buffer size

writes  $\leq$  PIPE\_BUF atomic

read blocks for 1 byte  
write blocks til complete

pipe EOF?

```
int main () {
    int pfd[2];
    int nread;
    char s[100];
    pipe(pfd);
    write(pfd[1], "hello", 6);
    nread = read(pfd[0], s, sizeof(s));
    printf("read %d bytes: %s\n", nread, s);
}
```

```
$ ./pipetest1
read 6 bytes: hello
```

Shell “pipe” facility

*\$ last*

```
lee pts/1 64.131.105.88 Mon Oct 5 12:50 still logged in
gport pts/1 sb112fh22.novell Mon Oct 5 12:46 - 12:49 (00:02)
ugandhi pts/1 216.47.150.142 Mon Oct 5 12:35 - 12:35 (00:00)
bhinshaw pts/1 dhcp18.northsout Mon Oct 5 12:16 - 12:23 (00:06)
gzajac1 pts/4 64.131.103.131 Mon Oct 5 11:57 - 12:07 (00:09)
sjordan1 pts/1 64.131.105.163 Mon Oct 5 11:39 - 12:05 (00:26)
sjordan1 pts/1 64.131.105.163 Mon Oct 5 11:38 - 11:39 (00:00)
neskey pts/3 64.131.105.81 Mon Oct 5 11:26 - 12:10 (00:44)
aciarkow pts/2 dhcp187.eastfowl Mon Oct 5 10:58 still logged in
aciarkow pts/5 dhcp187.eastfowl Mon Oct 5 10:35 - 10:57 (00:22)
aperezd pts/4 64.131.102.147 Mon Oct 5 10:28 - 11:15 (00:47)
tcraig pts/5 64.131.105.233 Mon Oct 5 10:22 - 10:27 (00:05)
aperezd pts/4 64.131.102.147 Mon Oct 5 10:10 - 10:27 (00:17)
marroyo1 pts/3 netserv9.gl.iit. Mon Oct 5 09:58 - 10:59 (01:01)
sdasari2 pts/2 host223.etc.iit. Mon Oct 5 09:56 - 10:49 (00:53)
sdasari2 pts/2 64.131.103.175 Mon Oct 5 09:48 - 09:53 (00:04)
sdasari2 pts/1 64.131.103.175 Mon Oct 5 09:22 - 11:36 (02:13)
amis pts/0 gamma15.cs.iit.e Mon Oct 5 08:51 still logged in
$
```

```
$ last | uniq -w 10
```

```
lee pts/1 64.131.105.88 Mon Oct 5 12:50 still logged in
gport pts/1 sb112fh22.novell Mon Oct 5 12:46 - 12:49 (00:02)
ugandhi pts/1 216.47.150.142 Mon Oct 5 12:35 - 12:35 (00:00)
bhinshaw pts/1 dhcp18.northsout Mon Oct 5 12:16 - 12:23 (00:06)
gzajac1 pts/4 64.131.103.131 Mon Oct 5 11:57 - 12:07 (00:09)
sjordan1 pts/1 64.131.105.163 Mon Oct 5 11:39 - 12:05 (00:26)
neskey pts/3 64.131.105.81 Mon Oct 5 11:26 - 12:10 (00:44)
aciarkow pts/2 dhcp187.eastfowl Mon Oct 5 10:58 still logged in
aperezd pts/4 64.131.102.147 Mon Oct 5 10:28 - 11:15 (00:47)
tcraig pts/5 64.131.105.233 Mon Oct 5 10:22 - 10:27 (00:05)
aperezd pts/4 64.131.102.147 Mon Oct 5 10:10 - 10:27 (00:17)
marroyo1 pts/3 netserv9.gl.iit. Mon Oct 5 09:58 - 10:59 (01:01)
sdasari2 pts/2 host223.etc.iit. Mon Oct 5 09:56 - 10:49 (00:53)
amis pts/0 gamma15.cs.iit.e Mon Oct 5 08:51 still logged in
lee pts/1 64.131.105.88 Mon Oct 5 08:43 - 08:46 (00:02)
amis pts/0 gamma15.cs.iit.e Mon Oct 5 08:41 - 08:51 (00:09)
iieong pts/1 adsl-76-197-134- Sun Oct 4 22:52 - 01:06 (02:13)
koleary2 pts/1 c-67-167-36-178. Sun Oct 4 22:08 - 22:47 (00:38)
$
```

```
$ last | uniq -w 10 | head -10
lee      pts/1      64.131.105.88    Mon Oct 5 12:50    still logged in
gport    pts/1      sb112fh22.novell Mon Oct 5 12:46 - 12:49    (00:02)
ugandhi  pts/1      216.47.150.142  Mon Oct 5 12:35 - 12:35    (00:00)
bhinshaw pts/1      dhcp18.northsout Mon Oct 5 12:16 - 12:23    (00:06)
gzajac1  pts/4      64.131.103.131  Mon Oct 5 11:57 - 12:07    (00:09)
sjordan1 pts/1      64.131.105.163  Mon Oct 5 11:39 - 12:05    (00:26)
neskey   pts/3      64.131.105.81   Mon Oct 5 11:26 - 12:10    (00:44)
aciarkow pts/2      dhcp187.eastfowl Mon Oct 5 10:58    still logged in
aperezd  pts/4      64.131.102.147  Mon Oct 5 10:28 - 11:15    (00:47)
$
```

```
int main () {
    int pfd[2];
    pid_t pid1, pid2;
    pipe(pfd);
    if (pid1 = fork() == 0) {
        dup2(pfd[1], STDOUT_FILENO);
        execlp("who", "who", NULL);
    }
    if (pid2 = fork() == 0) {
        dup2(pfd[0], STDIN_FILENO);
        execlp("sort", "sort", NULL);
    }
    waitpid(pid1, NULL, 0);
    waitpid(pid2, NULL, 0);
}
```

```
$ ./pipetest2
```

(hangs)

```
...
if (pid1 = fork() == 0) {
    dup2(pfd[1], STDOUT_FILENO);
    close(pfd[0]);
    close(pfd[1]);
    execlp("who", "who", NULL);
}
if (pid2 = fork() == 0) {
    dup2(pfd[0], STDIN_FILENO);
    close(pfd[0]);
    close(pfd[1]);
    execlp("sort", "sort", NULL);
}
close(pfd[0]);
close(pfd[1]);
...
```

```
$ ./pipetest2
efarrar pts/10 Nov 21 22:42
jkwon7 pts/1 Nov 22 14:31
lee pts/3 Nov 23 01:17
root pts/20 Nov 21 14:28
$
```

pros/cons

# Named Pipes (FIFOs)

```
int mkfifo (const char* path,  
            mode_t perms)
```

filesystem as namespace

follow with opens

implicit synchronization

```
$ mkfifo /tmp/my.fifo
$ ls -l /tmp/my.fifo
prw----- 1 lee wheel 0 24 Apr 04:13 /tmp/my.fifo

$ cat /tmp/my.fifo &
[1]+ cat /tmp/my.fifo &

$ echo hello > /tmp/my.fifo
hello
[1]+ Done cat /tmp/my.fifo

$ cat /etc/passwd > /tmp/my.fifo &
[1]+ cat /etc/passwd >/tmp/my.fifo &

$ cat /tmp/my.fifo
root:*:0:0:System Administrator:/var/root:/bin/tcsh
lee:*:501:20:Michael Lee,,,:/Users/lee:/bin/bash
[1]+ Done cat /etc/passwd >/tmp/my.fifo

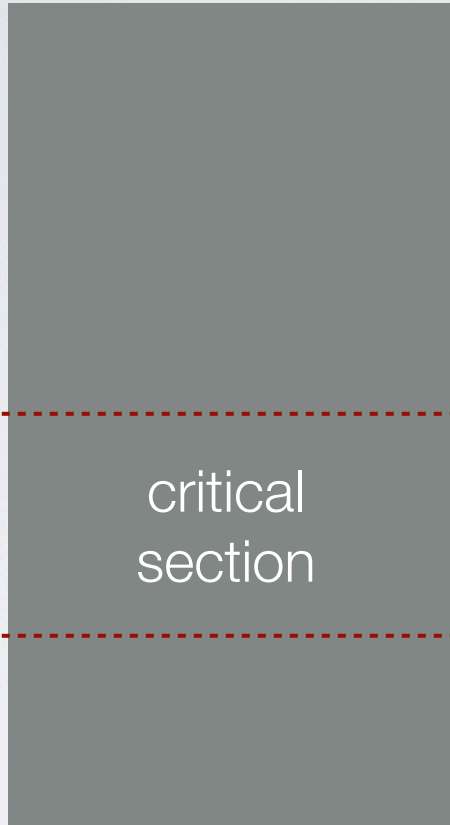
$ rm /tmp/my.fifo
```

# mkfifo Utility

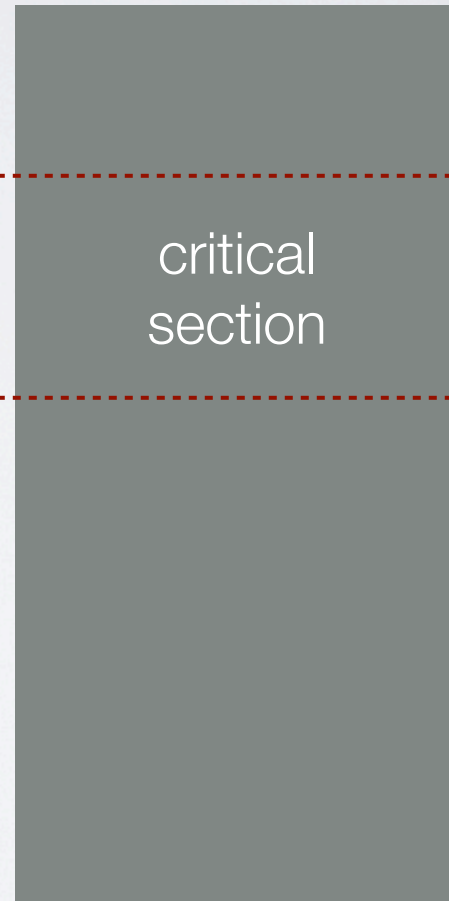
data sharing

synchronization

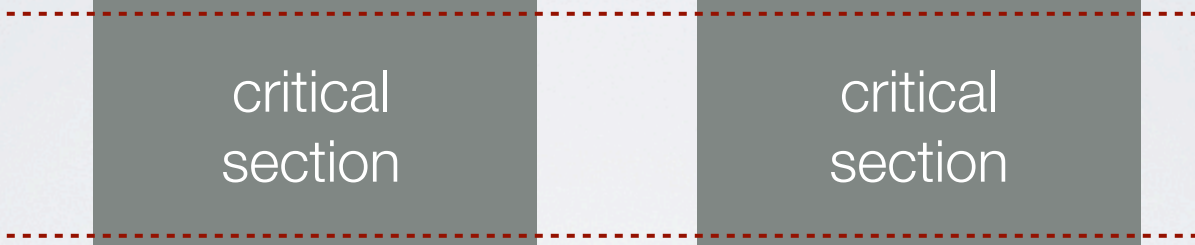
P<sub>1</sub>



P<sub>2</sub>



rendezvous



critical  
section

critical  
section

# File Locks

```
#include <fcntl.h>
```

```
int fcntl(int fildes,  
          int cmd, ...);
```

```
struct flock fl;
int fd;

fl.l_type    = F_WRLCK; /* F_RDLCK, F_WRLCK, F_UNLCK    */
fl.l_whence  = SEEK_SET; /* SEEK_SET, SEEK_CUR, SEEK_END */
fl.l_start   = 0;       /* Offset from l_whence        */
fl.l_len     = 0;       /* length, 0 = to EOF          */
fl.l_pid     = getpid(); /* our PID                     */

fd = open("filename", O_WRONLY); /* get the file descriptor */
fcntl(fd, F_SETLKW, &fl); /* set the lock, waiting if necessary */
.
.
.
fl.l_type = F_UNLCK; /* tell it to unlock the region */
fcntl(fd, F_SETLK, &fl); /* set the region to unlocked */
```

inter- vs. intra- system IPC

additional issues

transmission mechanism

routing

out-of-order data

architectural differences  
(e.g., endian-ness)