

Project B (CSE 422S)

Ken Wong
Washington University

kenw@wustl.edu
www.arl.wustl.edu/~kenw

Features/Issues

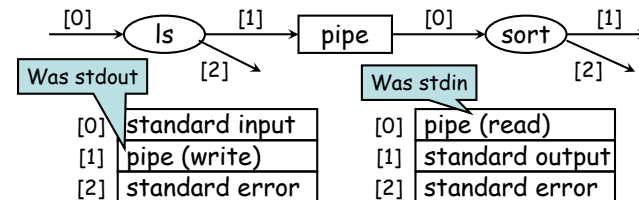
- Pipeline in foreground
- Pipeline in background
- Stdin/stdout redirection
- Handling ctrl-c from keyboard
 - » Should kill all processes in a foreground pipeline
 - `ls -l | sort -n +4 | more`
 - » What about background pipelines ???
- Background process termination notice
- Miscellaneous
 - » Parsing pipeline
 - » Check `#args` to built-in commands

Web Resources

- test40.C
 - » Shows effect of handling or ignoring SIGINT (ctrl-c)
- doio.c
 - » Shows: read stdin/write stdout using unbuffered I/O
- Project A command parser
 - » Can be used to parse one command (no pipe)
 - » But uses STL 'map'
- Homework 5, Problem 6 solution (To Come)
 - » Shows how to separate plumbing and I/O
 - But different structure than Project B
 - » `do_plumbing(); do_io();`
 - `do_plumbing: for (int i=0; i<npipes; i++) {Pipe(fd); ... }`
 - `do_io: while (Read(STDIN_FILENO, ...) > 0) Write(STDOUT_FILENO, ...);`

Pipeline (Review)

- Example (sort by non-decreasing file size)
 - » `ls -l | sort -n +4`



File Descriptor Tables

- Needed:
 - » Create pipe
 - » Replace STDIN/STDOUT with end of a pipe

Pipeline Movie (1)

xssh child:	[0] stdin		
	[1] stdout		
	[2] stderr		
xssh child: pipe	[0] stdin		
	[1] stdout		
	[2] stderr		
	fd[0] = [3] pipe.I		
	fd[1] = [4] pipe.O		
xssh child: fork	[0] stdin	[0] stdin	
	[1] stdout	[1] stdout	
	[2] stderr	[2] stderr	
	[3] pipe.I	[3] pipe.I	
	[4] pipe.O	[4] pipe.O	

5 - Ken Wong, 4/5/2006

Washington University in St. Louis

Pipeline Movie (2)

xssh child:	dup2(fd[1], X);	[0] stdin	[0] pipe.I
	close(fd[0]);	[1] pipe.O	[1] stdout
	close(fd[1]);	[2] stderr	[2] stderr
child:	dup2(fd[0], Y);	[3] 0	[3] 0
	close(fd[0]);	[4] 0	[4] 0
	close(fd[1]);		

where X is STDOUT_FILENO
Y is STDIN_FILENO

child

Very Important:
Close both ends of all pipes

Aside:
If xssh creates more than one pipe, close both ends of ALL pipes before using them

6 - Ken Wong, 4/5/2006

Washington University in St. Louis

Foreground Pipeline

- Parsing
 - » "ls -l /usr/bin | sort -n +4 | more" becomes
 - » "ls -l /usr/bin", "sort -n +4", "more"
 - i.e., 3 strings, each representing a command
 - We know how to handle individual commands
- Approach
 - » xssh sets up pipes for 3 children and fork-execs the 3 processes for 'ls', 'sort' and 'more'
 - » Creates 4 processes in the same *process group*
 - xssh is the process group leader; i.e., process group = pid of xssh
 - » ctrl-c: SIGINT sent to and terminates all 4 processes !!!
- Need To:
 - » Put the 3 children in a new process group
 - » xssh should catch SIGINT; send SIGINT to processes

7 - Ken Wong, 4/5/2006

Washington University in St. Louis

Process Group Example

- script0


```
#!/bin/sh
echo "pid of this shell is $$"
ls -l | sort -n +4 | mysleep $1 &
ps -o pid,ppid,pgrp,cmd
```

sh script that calls sleep command
- script0 Output from "script0 60"


```
pid of this shell is 27129
PID PPID PGRP CMD
17233 17232 17233 -tcsh
27129 17233 27129 /bin/sh ./script0 60
27131 27129 27129 sort -n +4
27132 27129 27129 /bin/sh ./mysleep 60
27133 27129 27129 ps -o pid,ppid,pgrp,cmd
27134 27132 27129 sleep 60
```

my shell

-KILL
- "kill -9 -27129" kills all processes in group 27129

8 - Ken Wong, 4/5/2006

Washington University in St. Louis

Process Group

- Def. A collection of processes that can receive signals from the same terminal
- Every process belongs to some process group
- A *process group leader* is a process whose PID is the same as its PGRP
 - » The leader can die or join another group
- A child created by `fork()` inherits the PGRP of the parent
- Some functions
 - » `pid_t getpgrp(void)`
 - Return PGRP of caller
 - » `int setpgid(pid_t pid, pid_t pgid)`
 - Change PGRP of child process 'pid' (current process if 0)

Beware of race conditions !!!

9 - Ken Wong, 4/5/2006

Washington University in St. Louis

xssh

- Is in same process group as the shell that exec'd it
- Should ignore SIGINT
 - » `ctrl-c` will terminate it's foreground children but not xssh
 - » Won't terminate xssh's parent (login shell) either
 - Shell's ignore SIGINT
- Change process group of background pipelines
 - » `ctrl-c` will have no effect on these processes

10 - Ken Wong, 4/5/2006

Washington University in St. Louis