Chapter 13		
Shell IPC		
Overview		
Overview		
Co-processes		
• Signals		
Named Pipes		
• The wait Command		
	J	
	•	
Lesson: Co-processes		
	•	
What is Co-process Exchanging information		
Exchanging mornation		

What	is	Co-	pro	cess

(asynchronous information wi	a way to run commands in parallel sly) and keep a way to exchange th them. Korn shell family
command &	Execute command as co-process
communa G	Execute confinant as co-process
<i>n</i> <&p	Redirect co-process input to file descriptor n (by default stdin)
	Redirect co-process input to file descriptor n (by default
<i>n</i> <&p	Redirect co-process input to file descriptor n (by default stdin) Redirect co-process output to file descriptor n (by default

Exchanging information

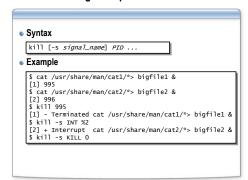
- Korn shell family has alternate to echo command print
- Both print and read commands can interact with coprocess.
- Example:

bc |& print -p "3*4" read -p answer echo \$answer

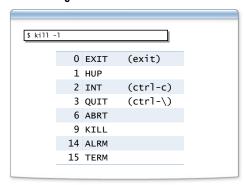
Lesson: Signals

- Sending signals
- Common signals
- The trap Command

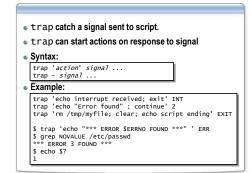
kill - send signal to processes



Common Signals



The trap Command



ı	_esson:	N	lan	har	Ρi	nec
ı	Lesson.	· I	ıan	ıeu	ГI	ue:

Named pipe is special file that allows two processes communicate with each other.

mkfifo named_pipe
[me@linux]~# mknod named_pipe p

After creating, processes can read from and write to that file.



Lesson: wait - await process completion

• Syntax:

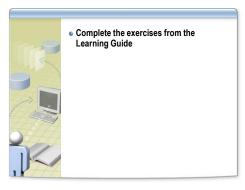
wait [job_id ...]

• Example:

\$ sleep 1000 & \$ pid=\$!

\$ wait \$pid ; echo Job exited with status \$?

Review Exercises



Topics for Review

1 Read the review topics 2 Think about what you learned in this Session in the context of your own work environment 3 Discuss your answers as a class	