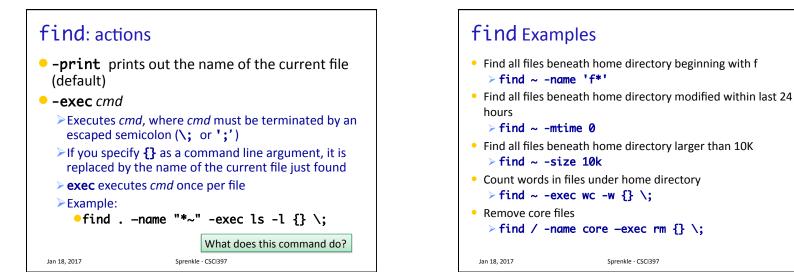
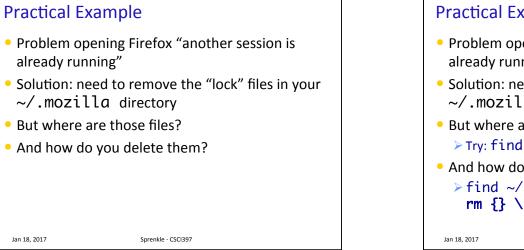


find: logical operations

Logical Operation	Functionality
! expression	returns the logical negation of expression
op1 -a op2	matches both patterns op1 and op2
op1 -o op2	matches either op1 or op2
()	group expressions together
Jan 18, 2017	Sprenkle - CSCI397





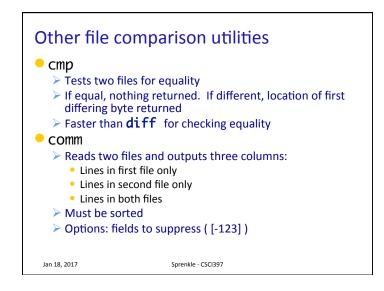
Practical Example

- Problem opening Firefox "another session is already running"
- Solution: need to remove the "lock" files in your ~/.mozilla directory
- But where are those files? >Try: find ~/.mozilla -name "*lock*"
- And how do you delete them? > find ~/.mozilla -name "*lock" -exec rm {} \;

Sprenkle - CSCI397

```
diff: comparing two files
• diff: compares two files and outputs a description of
  their differences
   > Usage: diff [options] file1 file2
   > -i : ignore case
   > -u : human readable
   ➤ -bB : ignore white space
                    apples
                               apples
                    oranges
                               oranges
                    walnuts
                               grapes
                    $ diff list1 list2
                    3c3
                    < walnuts
                    > grapes
 Jan 18, 2017
                         Sprenkle - CSCI397
```

Jan 18, 2017

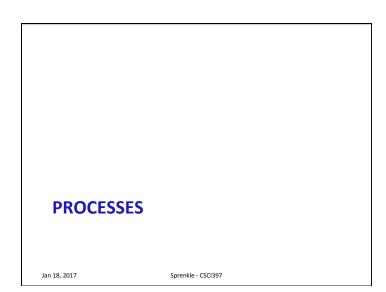


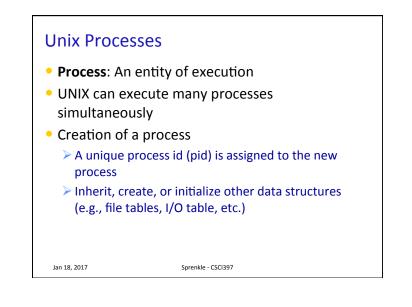


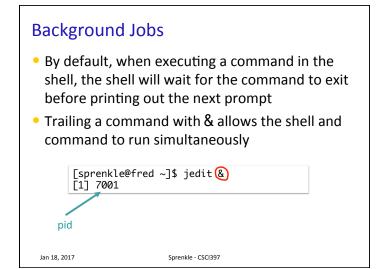
Сс	Control-Commands		
	Control +	Function	
	С	Interrupt or break job; stops printing and returns to UNIX	
	Z	Suspend current job bg to run in background	
	h	Erase or backspace character	
	S	Freezes screen	
	q	Unfreezes screen	
	u	Erase everything on line before this	
	w	Erase previous word	
	k	Erase remainder of line	

Sprenkle - CSCI397

Jan 18, 2017







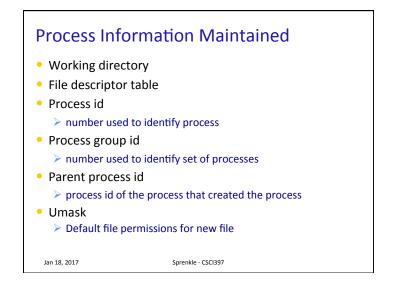
Ending a process

- When a process ends, there is a return code (an integer) associated with the process
 - > 0 means success

Jan 18, 2017

>>0 represent various kinds of failure, up to process

Sprenkle - CSCI397



Process Information Maintained We haven't talked about these yet: Effective user and group id The user and group this process is running with permissions as Real user and group id The user and group that invoked the process Environment variables

ps

- Report a snapshot of the current processes
- By default, just displays processes in the current terminal

> Columns by default: PID, TTY, TIME, and CMD

- Accepted options:
 - UNIX options, which may be grouped and must be preceded by a dash
 - BSD options, which may be grouped and must not be used with a dash
 - > GNU long options, which are preceded by two dashes

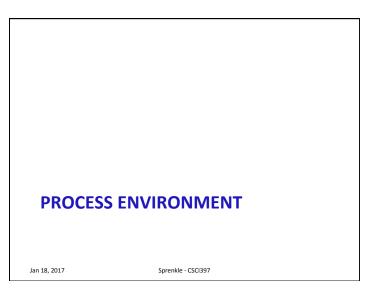
Jan 18, 2017

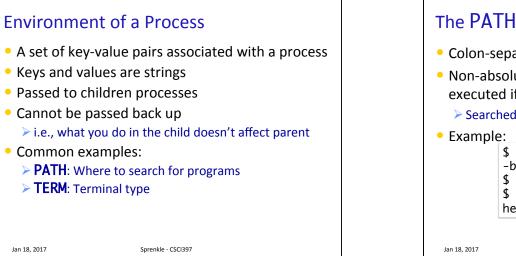
Sprenkle - CSCI397

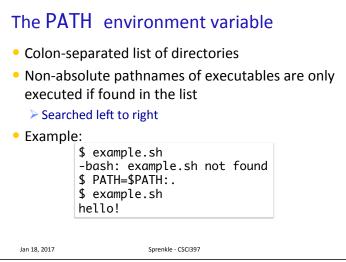
ps Examples

Command	Meaning
ps -e	See every process on the system
ps -ef	See every process on the system, in full listing
ps ax	See every process on the system
ps -ejH	See a process tree
Pipe through more	

Utility	Functionality
top	Monitors tasks
kill <pid></pid>	Terminate a process Use –9 if bugger won't die
nohup <cmd></cmd>	Makes a command immune to hangup and terminal signal
sleep <#>	Sleep in seconds
nice <cmd></cmd>	Run processes at a low priority
lan 18, 2017	Sprenkle - CSCI397

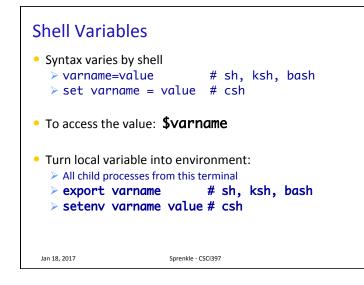






Having . In	Your Path	
\$ <i>ls</i> foo \$ <i>foo</i> sh: foo:	not found	\$. <i>/foo</i> Hello, foo.
and you ha	spath /	
Jan 18, 2017	Sprenkle -	CSCI397

Shell Variables	
 Shells have several mechanisms for creating 	
variables	
 A variable is a name representing a string value. Example: PATH 	
\succ Shell variables can save time and reduce typing errors	
 Allow you to store and manipulate information Ex: ls \$DIR > \$FILE 	
 Two types: local and environmental 	
Local are set by the user or by the shell itself	
Environmental come from the operating system and are passed to children	
Jan 18, 2017 Sprenkle - CSCI397	



Name	Meaning
\$HOME	Absolute pathname of your home directory
\$PATH	A list of directories to search for
\$MAIL	Absolute pathname to mailbox
\$USER	Your user name
\$SHELL	Absolute pathname of login shell
\$TERM	Type of terminal
\$PS1	Prompt
To view <i>all</i> shell variables, set	
Jan 18, 2017	Sprenkle - CSCl397

Setting Environment Variables

- You can set environment variables in your ~/.bash_profile file
- Open ~/.bash_profile using jedit or emacs or some text editor
- Create a new variable:
 > CS397=/csdept/courses/cs397
- Export the variable
 > export CS397
- In terminal, run the SOURCE command to load your new profile

> source ~/.bash_profile

- Check that your new variable was created: > echo \$C\$397
- Use the variable
 cd \$C\$397

Jan 18, 2017 Sprenkle - CSCI397

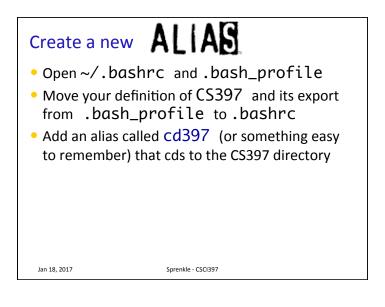
Bash's Configuration Files

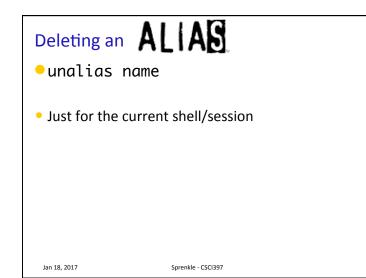
File Name	Purpose	
.bash_profile	Read and executed by Bash every time you log into the system	
.bashrc	Read and executed by Bash every time you start a subshell	
.bash_logout	Read and executed every time a login shell exits	
Open your .bash* files in jedit Notice what each file contains		
Jan 18, 2017	Sprenkle - CSCI397	

ALIAS

- Allow you to rename commands or type something simple instead of a list of options
- Can be defined on the command line, in .bash_profile, or in .bashrc
- To see all defined aliases
 > alias
- To see the definition for an alias
 > alias name
- To create an alias
 - >alias name=command

Jan 18, 2017





Sprenkle - CSCI397

