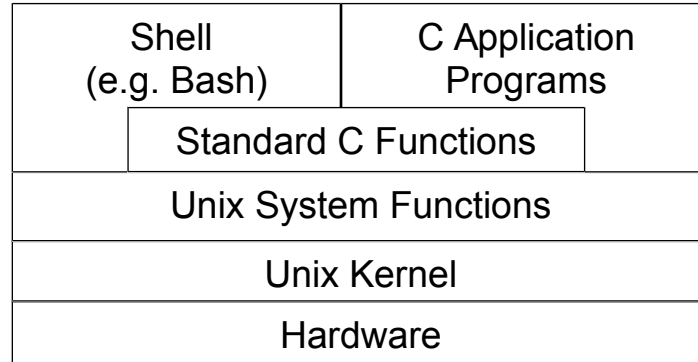


Princeton University

COS 217: Introduction to Programming Systems

Unix and Bash



File Names and Directory Names	
<i>/dir1/.../dirN</i>	Absolute dname
<i>dir1/.../dirN</i>	Relative dname
<i>/dir1/.../file</i>	Absolute fname
<i>dir1/.../file</i>	Relative fname

Special File Name and Directory Name Characters	
<i>fnameord*name</i>	* matches 0 or more characters
<i>fnameord?name</i>	? matches any single character
<i>"fname or dname"</i>	" allows whitespace in a dname or fname
<i>'fname or dname'</i>	' allows whitespace in a dname or fname
<i>fnameord\<i>name</i></i>	Backslash (escape) character allows special characters in a dname or fname
<i>~loginid</i>	Home directory of <i>loginid</i>
~	Your home directory
..	Parent of working directory
.	Working directory

Command for Getting Help	
man [<i>section</i>] <i>pagename</i>	(bin) Write to stdout the Unix manual page (from <i>section</i>) whose name is <i>pagename</i> . Section 1 describes commands and utilities (e.g. cat, ls). Section 2 describes Unix system functions (e.g. fork(), dup()). Section 3 describes library functions (e.g. printf(), strlen()).

Directory-Related Commands	
pwd	(Bash, bin) Write (print) the name of the working directory to stdout
cd [<i>dname</i>]	(Bash) Make <i>dname</i> the working directory
ls [-la] [<i>dname</i>]	(bin) List the contents of <i>dname</i> to stdout
ls [-la] [<i>fname</i>]	(bin) List the attributes of <i>fname</i> to stdout
mkdir <i>dname</i>	(bin) Create <i>dname</i>
rmdir <i>dname</i>	(bin) Destroy the empty directory <i>dname</i>

File-Related Commands	
cat	(bin) Concatenate (write) stdin to stdout
cat <i>fname</i> ...	(bin) Concatenate (write) <i>fname</i> ... to stdout
more <i>fname</i> ...	(bin) Write <i>fname</i> ... to stdout one screen at a time
less <i>fname</i> ...	(bin) Write <i>fname</i> , ... to stdout one screen at a time The man command pipes its output through less
xxd <i>fname</i>	(bin) Hexadecimal dump <i>fname</i> to stdout
cp [-i] <i>sourcefname targetfname</i>	(bin) Copy <i>sourcefname</i> to <i>targetfname</i>
cp [-i] <i>sourcefname targetdname</i>	(bin) Copy <i>sourcefname</i> to <i>targetdname</i>
cp -r <i>sourcedname targetdname</i>	(bin) Copy (recursively) <i>sourcedname</i> to <i>targetdname</i>
mv [-i] <i>sourcefname targetfname</i>	(bin) Rename <i>sourcefname</i> to <i>targetfname</i>
mv [-i] <i>sourcefname ... targetdname</i>	(bin) Move <i>sourcefname</i> ... to <i>targetdname</i>
rm [-i] <i>fname</i> ...	(bin) Remove <i>fname</i> ...
rm -r [-i] <i>dname [fname ...]</i>	(bin) Remove <i>dname</i> (recursively) and <i>fname</i> ...

Special Command Characters	
<i>command</i> 0< <i>fname</i> <i>command</i> < <i>fname</i>	Redirect stdin to <i>fname</i>
<i>command</i> 1> <i>fname</i> <i>command</i> > <i>fname</i>	Redirect stdout to <i>fname</i>
<i>command</i> 2> <i>fname</i>	Redirect stderr to <i>fname</i>
<i>command</i> 1> <i>fname</i> 2>&1	Redirect stdout and stderr to <i>fname</i>
<i>command</i> 1 <i>command</i> 2	Pipe from <i>command</i> 1 to <i>command</i> 2
^d	End of file
<i>command</i> &	Run <i>command</i> as a background process
^z	Turn my foreground process into a stopped background process
^c	Send a SIGINT signal
↑	Scroll backward through the command history list
↓	Scroll forward through the command history list
! <i>prefix</i>	Reissue the most recently issued command that begins with <i>prefix</i>
! <i>commandnum</i>	Reissue the command whose number is <i>commandnum</i> (see the "history" command)

Configuration Commands	
source <i>fname</i>	(Bash) Execute the shell script in <i>fname</i>
export <i>variable=value</i>	(Bash) Set environment <i>variable</i> to <i>value</i>
export PATH= <i>dname1:dname2:...</i>	(Bash) Set the PATH environment variable indicating that Bash should search <i>dname1</i> , <i>dname2</i> , ... to find commands that are specified as relative <i>fname</i> s
export MANPATH= <i>dname1:dname2:...</i>	(Bash) Set the MANPATH environment variable indicating that the man command should search <i>dname1</i> , <i>dname2</i> , ... to find man pages
<i>variable=value</i>	(Bash) Set shell <i>variable</i> to <i>value</i>
PS1=" <i>h:w\ \$</i> "	(Bash) Set the PS1 shell variable to indicate that the command prompt should contain the name of the host computer, a colon, the name of the working directory, a dollar sign, and a space
set -o <i>shelloption</i>	(Bash) Turn on <i>shelloption</i>
set +o <i>shelloption</i>	(Bash) Turn off <i>shelloption</i>
set -o ignoreeof	(Bash) Turn on the ignoreeof shell option to indicate that ^D entered at the Bash prompt should not terminate Bash
set -o noclobber	(Bash) Turn on the noclobber shell option to indicate that Bash should not overwrite files via redirection
alias <i>aliasname=string</i>	(Bash) Create an alias definition such that <i>aliasname</i> as an abbreviation for <i>string</i>
unalias <i>aliasname</i>	(Bash) Destroy the alias definition that defines <i>aliasname</i>

File and Directory Permission Commands	
id	(bin) Write to stdout my login id and the group(s) to which I belong
chmod <i>mask fnameordname</i> ...	(bin) Set the permissions of <i>fnameordname</i> ... as indicated by <i>mask</i>
chmod {u,g,o,a}{+,-}{r,w,x} <i>fnameordname</i>	(bin) Set the permissions of <i>fnameordname</i> ... for its owner (u), group (g), other (o), or all (a) by adding (+) or removing (-) read (r), write (w), or execute (x) permissions
umask	(Bash) Write to stdout the default permissions used when creating new files and directories
umask <i>mask</i>	(Bash) Set the default permissions used when creating new files and directories as indicated by <i>mask</i>

Software Development Commands	
emacs	(bin) Create or edit a text file using the Emacs editor
gcc217	(bin) Preprocess, compile, assemble, and link a program using options appropriate for COS 217; a variant of gcc
gdb	(bin) Debug a program
make	(bin) Build a program
ar	(bin) Create an archive file containing object code
gprof	(bin) Analyze the performance of a program

Miscellaneous Commands	
history	(Bash) Write a numbered command history list to stdout
passwd <i>oldpassword</i>	(bin) Change my password from <i>oldpassword</i>
wc [<i>fname</i> ...]	(bin) Write a count of characters, words, and lines in <i>fname</i> ... (or stdin) to stdout
date	(bin) Write the date and time to stdout
printenv [<i>variable</i>]	(bin) Write the definition of environment <i>variable</i> (or of all environment variables) to stdout
echo [<i>arg</i> ...]	(Bash, bin) Write <i>arg</i> ... to stdout
who	(bin) Write information about current users to stdout
grep <i>pattern fname</i> ...	(bin) Write each line of <i>fname</i> that contains <i>pattern</i> to stdout
sort [<i>fname</i>]	(bin) Write each line of <i>fname</i> (or stdin) in lexicographic order to stdout
diff <i>fname1 fname2</i>	(bin) Write an indication of the differences between the contents of <i>fname1</i> and <i>fname2</i> to stdout
which <i>command</i>	(bin) Search PATH for <i>command</i> , and write the dname where it was found to stdout

Process Control Commands	
jobs	(Bash) Write the names and jobnums of my background processes to stdout
fg [% <i>jobnum</i>]	(Bash) Move my background process with the given <i>jobnum</i> to the foreground
bg [% <i>jobnum</i>]	(Bash) Turn my stopped background process into a running background process
kill [- <i>signal</i>] % <i>jobnum</i>	(Bash) Send <i>signal</i> to my background process with the given <i>jobnum</i>
ps	(bin) Display a list of my processes
kill [- <i>signal</i>] <i>pid</i>	(bin) Send <i>signal</i> to the process whose id is <i>pid</i>
exit	(Bash) Exit Bash
logout	(Bash) Exit Bash and the terminal session