

# CS-344 - Unix Operating System Fundamentals

Lecture 3  
Using Basic UNIX Utilities  
and  
Using Multiple Utilities in Scripts

---

---

---

---

---

---

---

---

Based on slides created by  
Dr. Bangalore for the  
Spring 2005 offering of  
the course

---

---

---

---

---

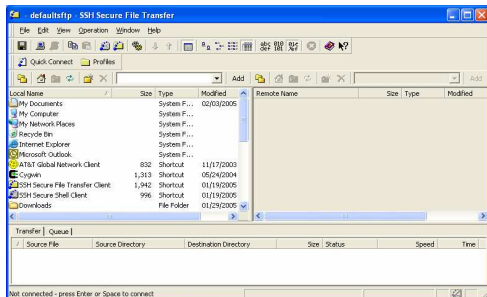
---

---

---

## Copying files to a UNIX system

Double-click on the SSH Secure File Transfer Client icon



9/1/2005

3

---

---

---

---

---

---

---

---





## UNIX Shell Unix Command Interpreter

- SHELL → Interface between the user and the UNIX Kernel
  - execute commands typed on the keyboard
  - prompt and waits for commands
  - interpret the commands and execute them
- UNIX commands can be internal/built in or external

9/1/2005

10

---

---

---

---

---

---

---

---

## Environment Variables

- Store information relevant to the shell or a particular program.
  - \$USER
  - \$LOGNAME
  - \$HOME
  - \$SHELL
  - \$PATH
  - \$CLASSPATH
- *set, env & printenv* (output varies depending on shell)
- Environment variables are usually give in uppercase

9/1/2005

11

---

---

---

---

---

---

---

---

## Shell, user-defined variables?

- Shell environment variables
  - Customize the environment in which the shell runs and are used for proper execution of the shell commands.
  - A copy of the variables is passed to every command that executes in the shell as its child.
  - Initialized at log on.
- User-defined variables
  - Used within shell scripts as temporary storage.
  - Can be made global and passed to the commands that executes in the shell script.
  - Need to be declared before they can be used.

9/1/2005

12

---

---

---

---

---

---

---

---

## Setting Environment Variables (BASH Shell)

- Temporary changes
  - Remains until logout or exit from the shell
    - `VARNAME=value_of_variable`
    - `export VARNAME`
  - Example: `EDITOR=vi; export EDITOR`
- Lasting changes
  - Enter change in `.bashrc` or `.bash_profile`
  - Save the file
  - Enter the command:
    - `source ~/.bashrc` or `source ~/.bash_profile`

9/1/2005

13

---

---

---

---

---

---

---

---

---

---

## Setting Environment Variables (C Shell)

- Temporary changes
  - Remains until logout or exit from the shell
    - `setenv VARNAME value_of_variable`
  - Example: `setenv EDITOR vi`
- Lasting changes
  - Enter change in `.cshrc`
  - Save the file
  - Enter the command:
    - `source ~/.cshrc`

9/1/2005

14

---

---

---

---

---

---

---

---

---

---

## Environment Variables (all sessions)

- To set an environment variable for all sessions:
  - edit the configuration file for the shell you're running
  - change the environment variable setting of interest
  - save the changes

Shell	Configuration File
Bourne (sh)	/etc/profile*, ~/.profile
Korn (ksh)	~/.kshrc
C (csh)	~/.login, ~/.cshrc
Bash (bash)	~/.bashrc, ~/.bash_profile
Z (zsh)	~/.zshrc
TC (tcsh)	~/.tcshrc

9/1/2005

15

---

---

---

---

---

---

---

---

---

---



## \$PATH

- Indicates the directories that a shell searches to find the file corresponding to an external command (or executable)

**For example:**

```
echo $PATH
/hf/sol8/grid/bin:./bin:/usr/bin:/sbin:/usr/sbin:/hf/sol8/eclipse:/hf/lang/SUNWspr/bin:/hf/lang/SUNWspr/lib:/hf/local/java/bin:/hf/sol8/SUNWns6:/hf/local/netscape4.76:/hf/local/bin:/hf/local/lib:/hf/local/latex2html:/hf/local/packages.d/aisiddin/gps5.50:/hf/local/lib/gcc-lib/sparc-sun-solaris2.6/egcs-2.91.66:/hf/local/lib/g++-include:/hf/local/xemacsc-21.1.6/src:/hf/ra/local/bin:/usr/dt/bin:/usr/dt/lib:/usr/openwin/bin:/usr/openwin/bin/xview
```

- bash → `/bin:/usr/bin`
- csh → `/bin /usr/bin`
- "which" or "whereis" display the location of the utilities
- ".", indicates the current directory

9/1/2005

19

---

---

---

---

---

---

---

---

---

---

## Shell Prompt

- To change the shell prompt:
  - For bash, type: `PS1='someprompt'`

```
puri@blazer1:--[501]$ PS1='[hostname: :date ] $ '
[blazer1:Sun Feb 13 19:14:06 CST 2005] $
```

- For csh, type: `set prompt='value'`

```
puri@blazer1:--[501]$ set prompt='[%m:%d %w %D %P] $ '
[blazer1:Sun Feb 13 19:14:06] $
```

- Use man tcsh or man bash to learn more about how to customize the prompt

9/1/2005

20

---

---

---

---

---

---

---

---

---

---

## Compressing and Archiving Files




---

---

---

---

---

---

---

---

---

---

## tar (tape archive)

- ❑ Originally designed to save file systems on tape as a backup.
- ❑ Commonly used to pack a directory hierarchy as an ordinary disk file.
- ❑ Creating tar files
  - `tar cvf archive_name.tar [files | directory]`
- ❑ List contents of tar file
  - `tar tvf archive_name.tar`
- ❑ Restoring archived files
  - `tar xvf archive_name.tar`

9/1/2005

22

---

---

---

---

---

---

---

---

## Other tar options

- ❑ `c`: create a new tape and archive files on it
- ❑ `r`: append files to a tar file
- ❑ `t`: list archive contents
- ❑ `u`: add files to tar file if not already in there or if they have been modified
- ❑ `x`: extract files from tar file (all files if not specified)
- ❑ `f`: use a file instead of a tape for archiving or restoring files
- ❑ `v`: use verbose mode
- ❑ `p`: preserve permissions
- ❑ `-`: output is sent to standard output
  - `tar cvf - `ls` > archive_name.tar`

9/1/2005

23

---

---

---

---

---

---

---

---

## gzip

- ❑ To compress files use "`gzip`"
  - `gzip filename` (a file `filename.gz` is created and original is removed)
- ❑ To expand a compressed file use "`gunzip`"
  - `gunzip filename.gz` (a file `filename` is created)
  - `gzip -d filename.gz`
- ❑ "`tar`" can be used to create a compressed archive, use "`tar cvfz archive_name.tar.gz files`"

```
$ tar cvfz examples.tar.gz examples
$ file examples.tar.gz
examples.tar.gz: gzip compressed data, from Unix
$ cp examples.tar.gz /tmp/
$ cd !$
$ gunzip -c examples.tar.gz | tar xvf -
```

9/1/2005

24

---

---

---

---

---

---

---

---

# Visual Editor

---

---

---

---

---

---

---

---

## Editors

- Editors can be used to create new files and edit existing files
- Popular UNIX Editors
  - vi or vim
  - emacs
  - pico
- Editors are programs like other utilities
- The shell launches the editor program and passes control to it, when the program terminates the shell gets back the control

9/1/2005

26

---

---

---

---

---

---

---

---

## Creating a new file with "vi"

- Usage: `vi <options> <filename>`
- Example:
  - `vi myfile` - open myfile
  - `vi +n myfile` - open myfile and move cursor to line n
  - `vi +/<string>` - open myfile and move cursor to the line with the first occurrence of "string"
- By default you are in "command" mode
- Type "i" to enter "insert" mode
- Type in the text
- Press ESC to exit "insert" mode
- Type ":w" to save file
- Type ":q" to quit
- You can also type ":wq" to save and quit

9/1/2005

27

---

---

---

---

---

---

---

---

## Inserting text

- To insert text into an existing file, enter one of the following commands (from the command mode) and enter the required text:
  - Insert at the current cursor position → press "i"
  - Insert after the current cursor position → press "a"
  - Insert at the start of the current line → press "I"
  - Insert at the end of the current line → press "A"
  - Insert a line above the current line → press "O"
  - Insert a line below the current line → press "o"
- All of the above commands will result in a mode change (from command to insert)
- Press ESC to exit the "insert" mode

9/1/2005

28

---

---

---

---

---

---

---

---

---

---

## Moving around (I)

- Right → l (→)
- Left → h (←)
- Up → k (↑)
- Down → j (↓)
- A number before the above commands will move the cursor by the number of characters specified (4h - move cursor 4 characters to the left from current position)
- To move to a specific line use: nG where n is the specified line number
- CTRL-G will indicate the current line number and the total number of lines in the file
- To move to a specific target use: /<targetstring> and press ENTER and press n to move to next occurrence of the "targetstring"

9/1/2005

29

---

---

---

---

---

---

---

---

---

---

## Moving around (II)

- Scroll
  - up by a page → CTRL-F
  - down by a page → CTRL-B
  - up by half a page → CTRL-U
  - down by half a page → CTRL-D
- Move to the
  - start of the current line → ^ or 0 (number zero)
  - end of the current line → \$
  - start of the previous line → - (minus sign)
  - start of the next line → +
- Move to the
  - start of next word → w (n words forward - nw)
  - start of previous word → b (n words backward - nb)
  - end of current word → e (end of n words - ne)

9/1/2005

30

---

---

---

---

---

---

---

---

---

---

## Other Commands

- Undo last text-change action → *u*
- Undo all changes made to the current line → *U* (assumes that you have not move the cursor from that line)
- Repeat last text-change action → *.* (period)
- Delete
  - Character(s) → *x* or *nx*
  - Word(s) → *dw* or *ndw*
  - Line(s) → *dd* or *ndd*
- Replace
  - Character → *r*
  - Word → *cw* (mode change)
  - Line (mode change)
    - *S* or *cc* (from the start of current line)
    - *R* or *C* (from current cursor position)

9/1/2005

31

---

---

---

---

---

---

---

---

---

---

## Using Pico (I)

- Usage: `pico <options> <filename>`
- Example: `pico myfile`
- Editing commands:
  - Move to end of line → *CTRL-E*
  - Move to beginning of line → *CTRL-A*
  - Cut/delete current line (also puts the line in a buffer so it can be undeleted/pasted) → *CTRL-K*
  - Undelete/paste the most recently cut line → *CTRL-U*
  - Mark a block of text → *CTRL-^* (hold down *CTRL*, *SHIFT* and the *6* key, then use the arrows to create the highlighted block)
  - Delete current character → *CTRL-D*
  - Justify/rejustify margin → *CTRL-J*
  - Insert existing file into file you're editing → *CTRL-R*

9/1/2005

32

---

---

---

---

---

---

---

---

---

---

## Using Pico (II)

- Save and exit:
  - Type *CTRL-X* – Pico will ask if you want to save changes
  - Press *y* – Pico will show the current file name
  - Press *ENTER* to keep current name, or, type in new name and press *ENTER*
- Exit without saving:
  - Type *CTRL-X* – Pico will ask if you want to save changes
  - Press *n*

9/1/2005

33

---

---

---

---

---

---

---

---

---

---

## Miscellaneous

---

- How to...
  - View information about your user id:
    - *whoami* (or *echo \$USER*)
  - Display the name of the machine that you are currently logged onto
    - *hostname*

---

---

---

---

---

---

---

---