

[Unix](#)[Directories](#)[Files](#)[Useful Commands](#)[Permissions](#)[tar](#)[More Commands](#)[Working Remotely](#)[Emacs](#)

Unix

L555

Dept. of Linguistics, Indiana University
Fall 2010

What is Unix?

Unix is an operating system, like DOS or Windows

- ▶ developed in 1969 by Bell Labs
- ▶ works well for single computers as well as for servers
- ▶ now also underlying operating system for Macs

Linux: open source version of unix (mostly for PCs)

Unix

Directories

Files

Useful Commands

Permissions

tar

More Commands

Working Remotely

Emacs

Directory Structure

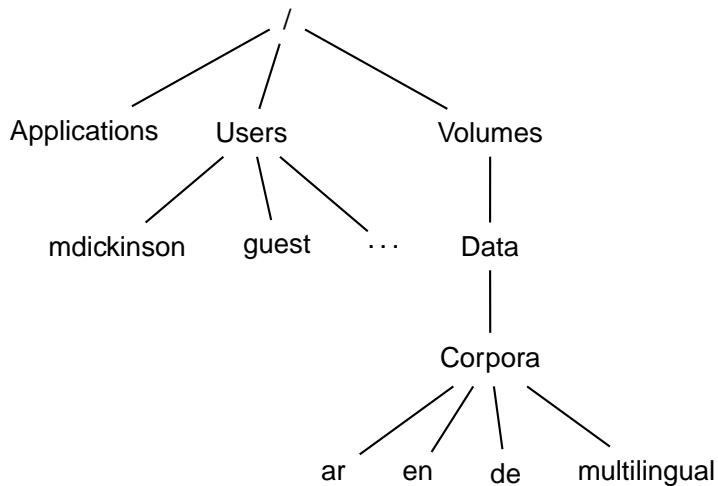
directory = folder

- ▶ directory structure: tree structure

Important directories:

- ▶ home directory: your private directory (/Users/guest)
- ▶ root node: /
- ▶ important directories on a Mac:
 - ▶ /Applications
 - ▶ /Users
 - ▶ /Volumes
 - ▶ /Volumes/Data/Corpora (on jones!)

Graphical Representation



Useful Commands for Directories

- ▶ list files in current directory:
`ls`
- ▶ list files with more information:
`ls -l`

Unix

Directories

Files

Useful Commands

Permissions

tar

More Commands

Working Remotely

Emacs

Useful Commands for Directories

- ▶ list files in current directory:
`ls`
- ▶ list files with more information:
`ls -l`
- ▶ change directories:
`cd <to>`
- ▶ two possibilities to specify <to>:
 - ▶ from root:
`cd /Users/mdickinson`

Unix

Directories

Files

Useful Commands

Permissions

tar

More Commands

Working Remotely

Emacs

Useful Commands for Directories

- ▶ list files in current directory:
`ls`
- ▶ list files with more information:
`ls -l`
- ▶ change directories:
`cd <to>`
- ▶ two possibilities to specify <to>:
 - ▶ from root:
`cd /Users/mdickinson`
 - ▶ from where we are:
`cd ../mdickinson`
.. means 'go up'

Useful Commands for Directories

- ▶ list files in current directory:
`ls`
- ▶ list files with more information:
`ls -l`
- ▶ change directories:
`cd <to>`
- ▶ two possibilities to specify <to>:
 - ▶ from root:
`cd /Users/mdickinson`
 - ▶ from where we are:
`cd ../mdickinson`
.. means 'go up'
- ▶ go back to home directory:
`cd`

Useful Commands for Directories

- ▶ list files in current directory:
`ls`
- ▶ list files with more information:
`ls -l`
- ▶ change directories:
`cd <to>`
- ▶ two possibilities to specify <to>:
 - ▶ from root:
`cd /Users/mdickinson`
 - ▶ from where we are:
`cd ../mdickinson`
.. means 'go up'
- ▶ go back to home directory:
`cd`
- ▶ show the directory where you are:
`pwd`

Useful Shortcuts

TAB completion

If you start typing a command or filename, then press TAB, the shell will complete the word for you – as far as possible.

Command history

The shell keeps a history of your commands. To scroll through them, simply press the up arrow key.

Looking at Files

- ▶ display contents of file in terminal:
`cat <filename>`

Looking at Files

- ▶ display contents of file in terminal:
`cat <filename>`
- ▶ display page by page:
`less <filename>`

Looking at Files

- ▶ display contents of file in terminal:
`cat <filename>`
- ▶ display page by page:
`less <filename>`
 - ▶ next page: space bar
 - ▶ quit: q

Looking at Files

- ▶ display contents of file in terminal:
`cat <filename>`
- ▶ display page by page:
`less <filename>`
 - ▶ next page: space bar
 - ▶ quit: `q`
 - ▶ go to beginning of file: `g`
 - ▶ go to end of file: `G`

Looking at Files

- ▶ display contents of file in terminal:

```
cat <filename>
```

- ▶ display page by page:

```
less <filename>
```

- ▶ next page: space bar
- ▶ quit: q
- ▶ go to beginning of file: g
- ▶ go to end of file: G
- ▶ search forward: /<expression> + hit return
- ▶ search backward: ?<expression> + hit return
- ▶ in search: next found occurrence: n

Looking at Files

- ▶ display contents of file in terminal:
`cat <filename>`
- ▶ display page by page:
`less <filename>`
 - ▶ next page: space bar
 - ▶ quit: `q`
 - ▶ go to beginning of file: `g`
 - ▶ go to end of file: `G`
 - ▶ search forward: `/<expression>` + hit return
 - ▶ search backward: `?<expression>` + hit return
 - ▶ in search: next found occurrence: `n`
- ▶ Likewise, there is the `more <filename>` command

Looking at Files

More Commands

- ▶ display beginning of file (normally 10):
`head <filename>`

Unix

Unix

Directories

Files

Useful Commands

Permissions

tar

More Commands

Working Remotely

Emacs

Looking at Files

More Commands

- ▶ display beginning of file (normally 10):
`head <filename>`
- ▶ display *X* beginning lines of file:
`head -n X <filename>`

Looking at Files

More Commands

- ▶ display beginning of file (normally 10):
`head <filename>`
- ▶ display *X* beginning lines of file:
`head -n X <filename>`
- ▶ display end of file (normally 10):
`tail <filename>`

Looking at Files

More Commands

- ▶ display beginning of file (normally 10):
`head <filename>`
- ▶ display *X* beginning lines of file:
`head -n X <filename>`
- ▶ display end of file (normally 10):
`tail <filename>`
- ▶ count number of lines, words, characters in a file:
`wc <filename>`

More Useful Commands

- ▶ list files that end in '.txt':

```
ls *.txt
```

Unix

Unix

Directories

Files

Useful Commands

Permissions

tar

More Commands

Working Remotely

Emacs

More Useful Commands

- ▶ list files that end in '.txt':
`ls *.txt`
- ▶ make directory:
`mkdir <path/><dirname>`

More Useful Commands

- ▶ list files that end in '.txt':

```
ls *.txt
```

- ▶ make directory:

```
mkdir <path/><dirname>
```

subdirectory: `mkdir results`

in your home directory: `mkdir ~/results`

More Useful Commands

- ▶ list files that end in '.txt':

```
ls *.txt
```

- ▶ make directory:

```
mkdir <path/><dirname>
```

```
subdirectory: mkdir results
```

```
in your home directory: mkdir ~/results
```

- ▶ copy a file from one location to another:

```
cp <filename> <path/>
```


More Useful Commands

- ▶ list files that end in '.txt':

```
ls *.txt
```

- ▶ make directory:

```
mkdir <path/><dirname>
```

```
subdirectory: mkdir results
```

```
in your home directory: mkdir ~/results
```

- ▶ copy a file from one location to another:

```
cp <filename> <path/>
```

- ▶ copy a file to the directory 'results' in your home directory:

```
cp dates.txt ~/results
```

More Useful Commands (2)

- ▶ sort your file:
`sort <filename>`

Unix

Unix

Directories

Files

Useful Commands

Permissions

tar

More Commands

Working Remotely

Emacs

More Useful Commands (2)

- ▶ sort your file:
`sort <filename>`
- ▶ sort (numbers) in numeric order:
`sort -n <filename>`

More Useful Commands (2)

- ▶ sort your file:
`sort <filename>`
- ▶ sort (numbers) in numeric order:
`sort -n <filename>`
- ▶ find a word in a file and display all the lines in which it occurred:
`grep <word> <fromfile>`

More Useful Commands (2)

- ▶ sort your file:
`sort <filename>`
- ▶ sort (numbers) in numeric order:
`sort -n <filename>`
- ▶ find a word in a file and display all the lines in which it occurred:
`grep <word> <fromfile>`
- ▶ find a word in a file and display all the lines in which it occurred and save results into a file in your home directory:
`grep <word> <fromfile> > ~/<tofile>`

More Useful Commands (2)

- ▶ sort your file:
`sort <filename>`
- ▶ sort (numbers) in numeric order:
`sort -n <filename>`
- ▶ find a word in a file and display all the lines in which it occurred:
`grep <word> <fromfile>`
- ▶ find a word in a file and display all the lines in which it occurred and save results into a file in your home directory:
`grep <word> <fromfile> > ~/<tofile>`
- ▶ example:
`grep linguistic mycorp.txt > ~/res.txt`

More Useful Commands (3)

- ▶ delete lines (adjacent) with identical content:
`uniq <filename>`

Unix

Unix

Directories

Files

Useful Commands

Permissions

tar

More Commands

Working Remotely

Emacs

More Useful Commands (3)

- ▶ delete lines (adjacent) with identical content:
`uniq <filename>`
- ▶ delete lines (adjacent) with identical content, but list how many were there:
`uniq -c <filename>`

More Useful Commands (3)

- ▶ delete lines (adjacent) with identical content:
`uniq <filename>`
- ▶ delete lines (adjacent) with identical content, but list how many were there:
`uniq -c <filename>`
- ▶ display differences between two files
`diff <filename1> <filename2>`

Permission System

▶ `ls -l`

```
-rw-r--r--    1 root   admin  168724 Nov  9  2003 Jokes.pdf
drwxr-xr-x   15 root   admin    510 Aug 31  2006 ar
drwxr-xr-x    3 root   admin    102 Aug  4  2004 cs
drwxr-xr-x    4 root   admin    136 Mar  2  2007 de
drwxr-xr-x   35 root   admin   1190 Feb 18 15:23 en
```

Unix

Unix

Directories

Files

Useful Commands

Permissions

tar

More Commands

Working Remotely

Emacs

Permission System

▶ `ls -l`

```
-rw-r--r--    1 root   admin 168724 Nov  9  2003 Jokes.pdf
drwxr-xr-x   15 root   admin   510 Aug 31  2006 ar
drwxr-xr-x    3 root   admin   102 Aug  4  2004 cs
drwxr-xr-x    4 root   admin   136 Mar  2  2007 de
drwxr-xr-x   35 root   admin  1190 Feb 18 15:23 en
```

▶ change permissions:

```
chmod <who><+/-><what> <file/dir>
```

<who>: user = u; group = g; others = o; all = a;

<what>: read = r; write = w; execute = x;

[Unix](#)[Directories](#)[Files](#)[Useful Commands](#)[Permissions](#)[tar](#)[More Commands](#)[Working Remotely](#)[Emacs](#)

Permission System

▶ `ls -l`

```
-rw-r--r--    1 root   admin  168724 Nov  9  2003 Jokes.pdf
drwxr-xr-x   15 root   admin    510 Aug 31  2006 ar
drwxr-xr-x    3 root   admin    102 Aug  4  2004 cs
drwxr-xr-x    4 root   admin    136 Mar  2  2007 de
drwxr-xr-x   35 root   admin   1190 Feb 18 15:23 en
```

▶ change permissions:

```
chmod <who><+/-><what> <file/dir>
```

<who>: user = u; group = g; others = o; all = a;

<what>: read = r; write = w; execute = x;

▶ example: give user write permissions for file 'hello.txt':

```
chmod u+w hello.txt
```

Permission System

▶ `ls -l`

```
-rw-r--r--    1 root   admin  168724 Nov  9  2003 Jokes.pdf
drwxr-xr-x   15 root   admin    510 Aug 31  2006 ar
drwxr-xr-x    3 root   admin    102 Aug  4  2004 cs
drwxr-xr-x    4 root   admin    136 Mar  2  2007 de
drwxr-xr-x   35 root   admin   1190 Feb 18 15:23 en
```

▶ change permissions:

```
chmod <who><+/-><what> <file/dir>
```

<who>: user = u; group = g; others = o; all = a;

<what>: read = r; write = w; execute = x;

▶ example: give user write permissions for file 'hello.txt':

```
chmod u+w hello.txt
```

▶ example: deny others write and execute permissions for directory 'test':

```
chmod o-wx test
```

Unix

Directories

Files

Useful Commands

Permissions

tar

More Commands

Working Remotely

Emacs

Packing and Unpacking Files

- ▶ pack a file:
`gzip <file>`
- ▶ results in `<file>.gz`

Unix

Unix

Directories

Files

Useful Commands

Permissions

tar

More Commands

Working Remotely

Emacs

Packing and Unpacking Files

- ▶ pack a file:
`gzip <file>`
- ▶ results in `<file>.gz`
- ▶ unpack file:
`gunzip <file>.gz`

Packing and Unpacking Files

- ▶ pack a file:
`gzip <file>`
- ▶ results in `<file>.gz`
- ▶ unpack file:
`gunzip <file>.gz`
- ▶ 'pack' more than one file:
`tar cvf <resultfile>.tar <file or directory>`
- ▶ unpack:
`tar xvf <file>.tar`

Packing and Unpacking Files

- ▶ pack a file:
`gzip <file>`
- ▶ results in `<file>.gz`
- ▶ unpack file:
`gunzip <file>.gz`
- ▶ 'pack' more than one file:
`tar cvf <resultfile>.tar <file or directory>`
- ▶ unpack:
`tar xvf <file>.tar`
- ▶ tar and gzip:
`tar cvzf <resultfile>.tgz <file|directory>`
- ▶ unpack:
`tar xvzf <file>.tgz`

More on Copy and Move

- ▶ copy a file from one location to another:
`cp <path/filename> <path/filename>`

Unix

Unix

Directories

Files

Useful Commands

Permissions

tar

More Commands

Working Remotely

Emacs

More on Copy and Move

- ▶ copy a file from one location to another:
`cp <path/filename> <path/filename>`
- ▶ move a file from one location to another:
`mv <filename> <path/>`

More on Copy and Move

- ▶ copy a file from one location to another:
`cp <path/filename> <path/filename>`
- ▶ move a file from one location to another:
`mv <filename> <path/>`
- ▶ rename a file:
`mv <oldfilename> <newfilename>`

More on Copy and Move

- ▶ copy a file from one location to another:
`cp <path/filename> <path/filename>`
- ▶ move a file from one location to another:
`mv <filename> <path/>`
- ▶ rename a file:
`mv <oldfilename> <newfilename>`
- ▶ remove a file:
`rm <filename>`

More on Copy and Move

- ▶ copy a file from one location to another:
`cp <path/filename> <path/filename>`
- ▶ move a file from one location to another:
`mv <filename> <path/>`
- ▶ rename a file:
`mv <oldfilename> <newfilename>`
- ▶ remove a file:
`rm <filename>`
- ▶ remove a directory including contents:
`rm -r <dir>`

Important Shortcuts in Paths

- ▶ `..` (go) up one directory
- ▶ `.` here
- ▶ `~` home directory

Unix

Unix

Directories

Files

Useful Commands

Permissions

tar

More Commands

Working Remotely

Emacs

Translate

- ▶ exchange characters for others:
`tr 'aeiou' 'X' < <filename>`
`tr 'aeiou' 'AEIOU' < <filename>`
- ▶ collapse characters (squeeze):
`tr -s 'J' < <filename>`
- ▶ exchange complement of characters for others:
`tr -c 'a-z' '\n' < <filename>`
`\n` is a linebreak

Putting Commands Together

- ▶ < use following filename as input file
- ▶ > use following filename as output file
- ▶ | use output of preceding command as input for following commands

Putting Commands Together

- ▶ `<` use following filename as input file
- ▶ `>` use following filename as output file
- ▶ `|` use output of preceding command as input for following commands

- ▶ example:

```
cat <infile> | tr 'A-Z' 'a-z' | sort >  
<outfile>
```

Find out More

```
man <command>
```

Unix

Unix

Directories

Files

Useful Commands

Permissions

tar

More Commands

Working Remotely

Emacs

Remote Login

- ▶ login:

```
ssh <username>@<machinename>
```

e.g. `ssh guest@jones.ling.indiana.edu`

- ▶ logout:

```
exit
```

Remote Copy

- ▶ to copy a local file to (your home directory) on another machine:

```
scp <filename> <machine>:
```

```
e.g. scp vm.pos jones.ling.indiana.edu:
```

- ▶ to copy a file from another machine to the current directory:

```
scp <machine>:<path>/<filename> .
```

```
e.g. scp jones.ling.indiana.edu:/Volumes/Data/  
Corpora/en/penntreebankv3/readme.all .
```

(in one line!)

Emacs

- ▶ emacs is a fairly basic text editor that can be run in a window or in the shell
- ▶ to start emacs:
`emacs <filename>`
- ▶ to quit:
`Ctrl-x Ctrl-c`
- ▶ save:
`Ctrl-x Ctrl-s`
- ▶ search:
`Ctrl-s`