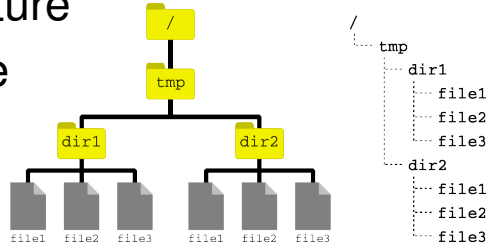# Information Literacy 07

commands

Ian Piumarta

Faculty of Engineering, KUAS

review

files and directories form a *tree* structure

the top is the *root* and it has no name

names that start with / are *absolute*

  • the *path* starts at the root directory

other names are *relative*

  • the path starts from the *current working directory*
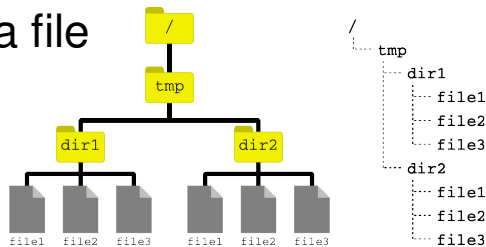
1

review

path elements are separated by the / character

each element names a directory

the final names either a directory or a file

```
/
/tmp
/tmp/dir1
/tmp/dir1/file{1,2,3}
/tmp/dir2
/tmp/dir2/file{1,2,3}
```



```
/
  tmp
    dir1
      file1
      file2
      file3
    dir2
      file1
      file2
      file3
```

review

every directory has entries for `.` and `..`

- `.` points to the directory itself
- `..` points to the *parent* directory

if the working directory is `/home/user` then

- `/home` and

- `..`

refer to the same directory

review

pwd prints the current working directory

cd changes your current working directory
- no argument: change to your home directory
- name as argument: change to that directory
- $-$ as argument: change to previous directory

review

`mkdir` *name*	creates a new directory
`cp` *file1 file2*	copies files
`cp` *files... dir*	copies files
`ls`	lists files/directories
`cat`	concatenates files
`grep`	searches for patterns
`wc`	counts words (lines, characters)
`cut`	prints fields from lines

review

`Control+C` terminates a running command

`Control+D` simulates end-of-file when reading keyboard

```
command > file
command < file
```
sends output to `file`
reads input from `file`

```
command1 | command2
```
send output of 'command1'
to input of 'command2'

`find` finds files and directories

```
find directory -property value
```

- `-type f` find regular files
- `-type d` find directories

default action is to print the name(s) found

Q.11 What command pipeline will count the number of directories under `/usr/lib` that have the digit `2` somewhere in their name?

mv moves (renames) files

```
$ mkdir temp
$ cd temp
$ touch file1
$ ls
file1
$ mv file1 file2
$ ls
file2
$ cd ..
$ rm -r temp
$ ls temp
ls: temp: No such file or directory
```

Q: 1

`curl` downloads files, `tar` extracts archives

```
$ curl -O https://kuas.org/tmp/metar-2019-10.tgz
  % Total    % Received % Xferd  Average Speed   Current
                                 Dload  Upload   Speed
100 1523k  100 1523k    0      0  3147k      0   3180k
```

```
$ tar -xzf metar-2019-10.tgz
```

- `-x` eXtract an archive
- `-z` decompress (unZip) the archive
- `-f` the Filename of the archive is given by the next argument

`less` shows files one page at a time

```
$ less metar-2019/2019-10-01T00:52:40-japan.txt
```

RETURN	to move one line down
SPACE	to move one page down
b	to move one page up
G	to move to the last page
g	to move back to the first page
q	to quit

head and tail show the ends of files

```
head file
```
- first 10 lines of *file*

```
head -n num file
```
- first *num* lines of *file*

```
tail file
```
- last 10 lines of *file*

```
tail -n num file
```
- last *num* lines of *file*

```
head -n p file | tail -n q
```

- last *q* lines of first *p* lines of *file*

- lines $p - q$ to p of *file*

`sort` sorts lines

`sort [`*`file`*`]`   ([x] means x is optional)

- sort lines alphabetically

`sort -n [`*`file`*`]`

- sort lines numerically

`sort -k` *n*

- sort lines on field *n* instead of entire line

`sort -t` *char* `-k` *n*

- use *char* to separate fields