1.27 Overview Solutions

1. *Changing password*

   (a) The system will tell you it is changing your LDAP password. It also tests the quality of your password using a library called `cracklib`, which rejects passwords that are too short or are dictionary words, or that are too closely related to your user name.

2. *Navigating Man Pages*

   (b) It is possible that some Linux distributions won't use `less` to display man pages. If that is the case, try to find out how you navigate under that setup and answer the same questions about it.

   (d) Keystrokes for basic man page navigation:

Instruction	Keystroke(s)
Top of man page	`g < ESC-<`
Bottom of man page	`G > ESC->`
Forward one screen	`f ^F ^V SPACE`
Backward one screen	`b ^B ESC-v`
Up one line	`y ^Y k ^K ^P`
Down one line	`e ^E j ^N RETURN`
pattern Search forward	/*pattern*
pattern Search backward	?*pattern*
Repeat *pattern* Search forward	`n`
Repeat *pattern* Search backward	`N`
Move to *n*th line	*n*g

Table 1.3: Keystrokes for basic man page navigation

N.B. Several different keystrokes can be used for the same movement. This is common in UNIX tools designed to operate from any keyboard. `less` always has a single key method. Multi-key methods are shown without spaces between them.

3. *Invoking the Right Man Pages*

   (a) **i.** `$` **`man -k whatis`**
   or, slightly differently:
   `$ man -f whatis`

   ii. `$ man -K cdrom` will print each man page name and ask you if you want to view the page or quit: `[ynq]`. If you just want the list of man pages and not view the man pages, you could answer "`n`" to each question using the `yes n` command, which just prints "`n`" to each prompt. You could do it like this:

   ```
   $ yes n | man -K cdrom
   ```

   . . . and if you wanted the names each on a line, with no prompt, then you could filter the output still further with the stream editor `sed`:

   ```
   $ yes n | man -K cdrom | sed 's/\? \[ynq\] /\n/g'
   ```

   but you would need to wait a while for the output.
   Suppose you wanted the output with just the name of the command without the directory name or the "`.gz`" at the end? Well, you could do this:

   ```
   $ yes n | man -K cdrom | sed 's/\? \[ynq\] /\n/g' | xargs --replace basename {} .gz
   ```

   And suppose. . . Oh well, never mind. Nick really gets carried away sometimes.

iii. There is no easy way to do this yet. Later on you will learn about `grep` which will allow you to filter the output of `man -k print` to see only the information you require. You could do it like this:

$ **man -k print | fgrep '(1)'**

(b) Practice using these flags to find and view man pages which deal with computer keywords your partner sets for you (and vice versa), e.g.

i. e.g. $ **man -K jpg**

ii. e.g. $ **man -K modem**

iii. e.g. $ **man -K NFS**

4. *Finding Out About Your System and Users*

(b) See Table 1.4.

Command string	Output
`$ whoami`	Your username
`$ who am i`	Your username plus machine(s) and terminal you are on
`$ users`	Usernames of currently logged on users
`$ who`	Who is logged on, when and where
`$ w`	Who's logged on, when, where, what process and what system resources they are using
`$ date`	Current date and time, can set date/time
`$ cal 10 2003`	Calendar for October 2003
`$ cal 9 1752`	Calendar for September 1752. Strange because 12 days were 'lost' in the transition from Gregorian to Julian calendars. Simply search for "September 1752" in Google
`$ df`	Disk free, i.e. summarises disk usage
`$ which man`	Full file and path name for the `man` executable file
`$ type man`	Much the same as `which man`
`$ whereis less`	Locates the `less` executable and its man page
`$ help cd`	Very brief help notes on the `cd` command. N.B. help only works on very few built-in commands
`$ time sleep 2`	The `sleep` command puts itself to sleep for 2 seconds. The `time` command then times the whole process and provides other data on the operation of the `sleep` command

Table 1.4: Output from basic system information commands

5. *Creating new files*

(a) Your output should be something like:

```
$ touch filename.txt
```

(b) Your output should be something like:

```
$ ls -l filename.txt
-rw-rw-r--   1 davef  davef  0 Jul 21 17:59 filename.txt
```

(c) Your output should now be something like:

```
$ touch filename.txt
$ ls -l filename.txt
-rw-rw-r--   1 davef  davef  0 Jul 21 17:59 filename.txt
$ touch filename.txt
$ ls -l filename.txt
-rw-rw-r--   1 davef  davef  0 Jul 21 18:01 filename.txt
```

i. The time stamp has changed

ii. The real purpose of touch is to change time stamps, but it is handy for creating new empty files

(d) **i.** Reading `diskspace.txt` should produce something like this:

```
$ cat diskpace.txt
Filesystem     Used Available Capacity Mounted on
/dev/hda1     65571   406394     14%   /
/dev/hdc1   5030416   650563     89%   /backup
/dev/hda5   2000097   857401     70%   /home
/dev/hda7     14289   457676      3%   /tmp
/dev/hda6   1136861   741727     61%   /usr
/dev/hdb     653004        0    100%   /mnt/cdrom
```

6. *Appending information to files*

(b) Your screen should look something like this:

```
$ w > test.txt
$ date >> test.txt
$ cat test.txt
  6:36pm  up 16 days, 23:07,  4 users,  load average: 1.03, 1.08, 1.02
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU  WHAT
davef    ttyp7    oakleigh:0.0      9:39am  0.00s  2.58s  0.02s  w
mikeb    ttyp4    kebab            Tue 3pm 23:07   0.20s  0.13s  -bash
davef    ttyp2    oakleigh          9:04am 15:34   7:24   0.07s  -bash
davef    ttypb    oakleigh          3:02pm  3:15m  3:00m  0.08s  -bash
Wed Jul 21 18:36:39 BST 1999
```

7. *Using Simple Pipes*

(a) `$ who | sort -r`

(b) `$ sort /etc/passwd > passwd`
or
`$ cat /etc/passwd | sort > passwd.sorted`

(c) `wc` prints the number of lines, words, and bytes in files. To get these details for your `/etc/mime.types` file, you could do the following:

```
$ cat /etc/mime.types | wc
    480     726   12682
```

i.e. 480 lines, 726 words, and 12682 bytes *
Another way of doing the same thing without a pipe is

```
$ wc /etc/mime.types
    480     726   12682 /etc/mime.types
```

(d) E.g.

```
$ cat /etc/mime.types | wc -w
    726
```

We can get a similar result by typing:
`$ wc -w /etc/mime.types`

*N.B. `wc` only counts whitespace-separated words