

APPENDIX B
A UNIX ENVIRONMENT (Partial)

Introduction

This appendix gives several examples of Things and their tasks in a Unix C-Shell Environment. I don't know if the current breakdown of each set of tasks into (1) tasks that give information, and (2) all other tasks, is one of the best, meaning one that would be preferred by a majority of Unix users. That could only be determined by testing on a large population of the users themselves. It is, at present, simply the breakdown that has proven most natural to me in the course of using Unix over several years.

I am sure there are omissions in each list of tasks, but let there be no doubt about the *goal*: for each Thing, *to list all the tasks that a user can perform on the Thing*. No user should ever have to *figure out* if a task he or she has in mind can be performed on a Thing. (I have excluded tasks reserved for the superuser from this appendix. In the actual Environment, of course, these are included, and clearly marked as such.)

The examples make the reasonable assumption that the Unix system has man (manual) pages, i.e., on-line information on each command (and a few other terms) which can be displayed by entering “man” followed by the name of the command or the term, e.g., `man rm`.

A *portion* of the Introduction to the Unix Environment (see Appendix A for a complete example) is also given, this portion providing a few more details about using the Things and the tasks listings.

Not all referenced sections are included in this appendix. Those not included are marked by “[not shown]” following the reference. A sequence of vertical dots denotes omitted material.

Unix Environment

.

.

.

Introduction

.

.

.

About Things and their Tasks

Things is a term for important Unix entities like files, directories, shells, scripts — yes, even users! Each such Thing has associated with it a set of tasks. In this Environment, *every* task which can be performed on each Thing is given under the name of the Thing. If you do not find the task you are looking for in the list of tasks, then it cannot be performed by a user for whom this Environment is intended!

The following conventions have been used in the lists of tasks:

- Commonly used forms of the command(s) implementing each task are given using normal syntax conventions (see above in this Introduction under “Conventions” [not shown]). For example, `cp <current file name> <new file name>`. If you need more information on the command, then simply enter `man <name of command>`.

- In cases where it was difficult to decide on a commonly used form, a reference to the man pages is given.

A word or phrase in double quotes means that the definition can be looked up if the user desires.

- In cases where an explanation of more than a line or two is required, a reference to the relevant topic in this Environment is given.

- At the end of each list of tasks, under “Related,” is a list of topics (not necessarily just the topics called Things) which are closely related to the Thing covered by the tasks.

.

.

.

Body of Environment

.

.

•
command

Definition of “command” in Unix:

See “command, definition of” [not shown].

Tasks to find information about commands

Given the name of a command, display information about the command syntax and semantics:

```
man <name of command>
```

Display a list of recently executed commands:

```
history
```

Display a list of the currently active abbreviations of commands and what they stand for:

```
more .cshrc
```

Other tasks on commands

Execute a command so that it runs in the “foreground:”

Following the prompt, type the command name, then press Enter.

Execute a command so that it runs in the “background:”

Same as for running command in “foreground,” except add a & before pressing Enter.

Stop a command from continuing execution, i.e., abort a command:

For a command executing in the foreground, press Ctrl-C.

For a command executing in the background, see under “process”: “Abort a process.”

Re-execute the previously entered command:

```
!!
```

For other options in executing previously-entered commands, see “command, re-execute a” [not shown].

Make a command begin execution after a specified time:

See “sleep” in man pages.

Make a command take its input from a file:

<command name> < <file name>

Put the output of a command into a file, erasing existing contents of file if file already exists:

<command name> > <file name>

For more details, see under “command, redirecting output of a” [not shown].

Make the output of a command be the input of the next command:

<command name> | <next command name>

For more details, see under “commands, pipelining of” [not shown].

Make a command continue executing even after you log out:

See “nohup” in man pages.

Change the priority of execution of a command:

See “nice” in man pages.

Abbreviate a (typically long) command:

alias <abbreviation> <command>

Related

See:

“event”

“process”

“script”

directory

Definition of “directory”:

See “directory, definition of” [not shown].

Tasks to find information about directories

Display the name of the current directory:

```
pwd
```

Display properties of current directory, including “permission”s:

```
ls -l -d
```

For explanation of results, see “ls” in man pages.

Display names of all directories below the current directory having a given partial “pathname:”

```
find . -name ' * <partial file name>* ' -type d -print
```

Other tasks on directories

Connect to a specified directory, i.e., make it the current directory:

```
cd <directory name>
```

Connect to your login name’s “home directory”:

```
cd
```

Create a directory:

```
mkdir <directory name>
```

Delete a directory:

1. Delete all files in the directory.

2. Enter:

```
rmdir <directory name>
```

Copy a directory to floppy disk(s):

See “bar”, “tar” commands in man pages.

Copy a directory from floppy disk(s) to hard disk:

See “bar”, “tar” commands in man pages.

Copy a directory to another directory:

See “mount” in man pages.

Change permissions on a directory:

See “chmod” in man pages.

Related

“file”

“file system” [not shown]

disk, hard

Definition of “hard disk”:

See “disk, hard, definition of” [not shown].

Tasks to find information about hard disks

Display a listing of id’s of all hard disks you currently have access to:

See “df” in man pages.

Find out memory usage on hard disk *x*

du

Other tasks on hard disks

Copy a file to floppy disk(s):

See “bar”, “tar” commands in man pages.

Copy a file from floppy disk(s) to hard disk:

See “bar”, “tar” commands in man pages.

Related

“main memory” [not shown]

“RAM” [not shown]

“ROM” [not shown]

file

Definition of “file”:

See “file, definition of” [not shown]:

Tasks to find information about files

List names of each file in current directory having string *x* in its name:

```
ls *x*
```

Display properties of a file:

```
ls -l <file name(s)>
```

Display type of a file:

See “file” in man pages.

Display permissions for a file named *x*:

```
ls -l x
```

Count number of lines, words, characters, in a file named *x*:

```
wc x
```

Display all lines in a file which contain a specified string:

```
grep <string> <file name>
```

See also “awk” in man pages.

Search for information, patterns in a file

See “grep”, “diff” in man pages.

Other tasks on files

Create a file:

See “cat” in man pages, or use editor, e.g., vi.

Copy a file within a given machine, server:

```
cp <current file name> <new file name>
```

Copy a file across a network:

See “cpio”, “ftp”, “bar”, “tar” in man pages.

Copy a file to, from, floppy disks:

For file(s) occupying more than one disk, use “bar” command. (See man pages.)

See also “tar” command in man pages.

Delete file(s):

```
rm <file name(s)>
```

Move file within a given machine, server:

```
mv <current file name> <new file name>
```

Concatenate two or more files into a new file which contains the originals in the order $file_1, file_2, file_3, \dots, file_n$:

```
cat file1,file2,file3, ...,filen > newfile
```

Merge two sorted files into a new, sorted file:

See “sort” in man pages.

Find a file:

E.g., find, in any directory in or below the present directory, all files whose name contains the string x , and print out the full name of each such file:

```
find . -name '* *x*' -print
```

Rename a file:

```
mv <current name of file> <new name of file>
```

Change permissions on a file:

```
chmod <new permissions mask> <file name>
```

Appendix B — A Unix Environment (Partial)

Display contents of a file named *x*:

Use `more x`, or `cat x`, or `head x`, or `tail x`
or use editor, e.g., `vi`.

Compare contents of two files:

See “diff”, “cmp”, “comm” in man pages.

Find a string or other pattern in a file:

```
grep <string> <file name>
```

See also “awk” in man pages.

Change contents of a file:

Use an editor, e.g., `vi`. See also “awk”, “sort” in man pages.

Compress a file.

See “compress” in man pages.

Related: “uncompress”.

Display classification of a file.

See “file” in man pages.

Display contents of a file.

See “edit a file” below.

Edit a file.

See “Emacs” [not shown], “FrameMaker” [not shown], “sed” [not shown],
“Text Editor” [not shown], “vi” [not shown].

Encode a binary file as ASCII for, e.g., transmission via e-mail

See “uuencode” in man pages.

.

Convert format of a file to Postscript format:

See “enscript” in man pages.

Establish file creation permissions mask:

See “umask” in man pages.

Change permissions on a file:

See “chmod” in man pages.

Compress a file.

See “compress” in man pages.

Related: “uncompress”.

Rename file.

`mv <current file> <file of new name>`

Modify time-related parameters of a file:

See “touch” in man pages.

Uncompress a file.

See “uncompress” in man pages.

Related: “Compress”

Related

“directory”

“file system” [not shown]

memory

See “CD-ROM” [not shown], “disk”, “main memory” [not shown], “RAM” [not shown], “ROM” [not shown], “floppy disk”.

process

Definition of “process”:

See “process, definition of” [not shown].

Appendix B — A Unix Environment (Partial)

Tasks to find information about processes

Get a list of all current processes:

```
ps -auxww
```

Other tasks on processes

Abort a process:

1. Find the “PID” of the process by entering:

```
ps
```

2. Kill the process by entering:

```
kill PID
```

Related

“command”

“event” [not shown]

“shell”

screen

Definition of “screen”:

See “screen, definition of” [not shown].

Other tasks on screens

Make a record of everything that appears on the screen:

Enter:

```
script <name of file in which to record screen contents>
```

To terminate record, press:

```
Ctrl-d
```

Erase screen

•

-
-
- Refresh screen
-
-
-

Related

“terminal”

“windowing systems” [not shown]

script

Definition of “script”:

See “script, definition of” [not shown].

Other tasks on scripts

Create a script:

See “script, create a”.

Edit a script:

Use any editor, e.g., vi.

View an existing script:

See “script, view a”.

Set execute permissions on a script:

See “script, set execute permissions on a” [not shown].

Run a script:

See “script, run a”.

Debug a script:

See “script, debug a”.

Related

“program”

shell

Definition of “shell”:

See “shell, definition of” [not shown].

Tasks to find information about shells

Display symbol identifying current shell:

ps

In result, “csh” stands for “C Shell” [not shown], “sh” stands for “Bourne Shell” [not shown], “ksh” stands for “Korn Shell” [not shown].

Other tasks on shells

Write a program using shell commands:

See “script”.

Related

“Bourne Shell” [not shown]

“C Shell” [not shown]

“Korn Shell” [not shown]

space

See “disk”, “main memory” [not shown], “RAM” [not shown], “ROM” [not shown].

terminal

Definition of “terminal”

See “terminal, definition of” [not shown].

Tasks to find information about terminals

Display a list of names or id’s of all terminals on system:

.
. .

Find out who owns a given terminal:

.
. .

Find out what terminals are assigned to a given user:

.
. .

Find out name, id, properties of your current terminal

.
. .

Sign on to another user’s terminal:

.
. .

Related

“screen”

“workstation” [not shown]

user

Definition of “user”, i.e., assumed minimum vocabulary and skills of any user of this Unix Environment:

See “user, definition of”.

Tasks to find information about users

Display list of names of other users

.
. .
.

Other tasks on users

Send message to other user(s):

See “e-mail” [not shown], “mail” [not shown].

Make up a mailing list of users:

See “alias” in man pages.

Sign on to another user’s terminal

`rlogin <name of user’s terminal>`

Related

“superuser” [not shown]

“system administrator” [not shown]

variable, Environment

Definition of Environment variable:

See “variable, Environment, definition of” [not shown].

Tasks to find information about Environment variables

List all current Environment variables:

`printenv`

Other tasks on Environment variables

Create an Environment variable

```
setenv <Environment variable name> <value>
```

Change the value of an Environment variable:

```
setenv <Environment variable name> <value>
```

Make recently entered values of Environment variables take effect throughout current shell:

```
source
```

Related

“script”

“script variables”

“shell”