Exercises-Man

In this exercise set we will investigate the use of the man system on linux.

You should be logged onto one of our lab or classroom linux systems for each of these exercises.

Exercise 1 - helpful system pages

There are several pages available that give you general information about the system. We will look at them one at a time:

1. Since man sections vary between implementations (after sections 1-3), probably the best reference would be a list of the sections and what they contain. I wish this was standard. On linux this can be found in man(1) (yes, man man). Examine man man on linux.
2. Check out man hier. Even though this is in different sections on the two systems, both give you information about the organization of standard directories.
3. The intro page of each section tells you about that section. Thus if you want to find out what section 4 is about use man 4 intro.
4. man fs tells you about the various types of filesystems. Try it.
5. man proc tells you about the /proc filesystem. We will investigate the /proc filesystem later.
6. To get you started in system administration, check out man bootup. This is a good introduction to a lot of stuff we will be learning later.

Exercise 2 - manual page sections

1. Sections 1-3 of the Unix manual are the sections standardized for user information: 1 describes commands, 2 describes system calls, and 3 describes library functions. The other sections, which describe more system information interesting to us, are not standard. You can investigate what a particular section is on by reading the intro page of that section using man N intro, where N is the section you are interested in.
2. Investigate the intro section for sections 4, 5, 6 and 7 on linux.
3. Section 7 is often of interest to administrators. Unfortunately, other sections sometimes get in the way. For example, time(7) describes how time is measured on linux, but time(1) is searched first. If you are interested in looking up administrative topics, you may have to modify how man searches sections to find the information you want. There are several ways to do this:
   - You could add the -a option to man to output all pages rather than just the first one found. Try man -a time. (Several pages will display in succession. To skip to the next page type q.)
   - You could limit your search to section 7 using man 7 topic. This means you will not find information in the other sections that might have relevance such as 4 and 5.
   - You could change the section order using your own section list. The first way to do this is man -S 7:4:5 topic. Try it with time.
   - The second way to change your section order is to define your own MANSECT environment variable. Define MANSECT as 7:4:5. (Don't forget to export it.) Try it with time again. Don't forget to unset it when you are done.

Exercise 3 - man trees

The basic unit of data for the man system is a man tree. A man tree has a base (root) directory, usually named man. (If it is not named man you will have to define your MANPATH by hand.) In this directory there must be subdirectories named man1, man2, ... for each manual section that has pages in this tree. In each subdirectory are the pages, possibly gzipped. Pages are named for their topic with the suffix .N where N is the section. Thus the page for the command find is in the standard tree at /usr/share/man/man1/find.1.gz

1. Check out the tree /usr/local/share/man. List it recursively. This [empty] tree is for the addition of local pages. Note that it is usually searched after the standard tree, so it cannot be used to supersede a standard page (see below).
2. Check out the tree /usr/java/latest/man. The only standard subdirectory we see is man1. (We will...
discuss the other ones later.) List it. These are the java man pages corresponding to the current default version of java on our machines. Are they compressed?

3. One of the pages in /usr/java/latest/man/man1 is javac.1. There is a version of this page in the standard man tree. Can you find it?

4. Looking at /usr/java/latest/man again, the other subdirectories are language-specific. List one of them. You should find it is another man tree.

**Exercise 4 - MANPATH**

Just like your PATH determines the order that commands are found, your MANPATH sets the order that your man trees are found. You can set your own MANPATH. In the absence of a MANPATH environment variable, the path man uses is created for you on-the-fly using several pieces of information:

- the man configuration file /etc/man.config (RH6) or /etc/man_db.conf (RH7)
- your PATH
- your locale

We are going to investigate each of these in turn to dispel some of the confusion.

Note: for the rest of these exercises, when you are asked to examine your MANPATH, it means using man --path unless you have set your own MANPATH variable. (This is a bit of a misnomer - man --path doesn't actually create a MANPATH variable - it just simulates it on the fly.)

1. Start a new shell by typing bash. Then delete (using unset) your environment variables PATH and LANG. The purpose of this is to see the default MANPATH with no modifications due to your PATH or LANG.

2. Run the command /usr/bin/man --path. (You could also use the command /usr/bin/manpath, which may be a link to /usr/bin/man.) This shows your MANPATH as configured by default by /etc/man_db.conf. Write down this default MANPATH. Then run man on nmap. Note the language it displays in.

3. Your default MANPATH should have the standard tree and the local man tree. In addition it should have one curious item: the overrides directory. This is added to the standard man tree to allow some standard man pages to be superseded. The way to think about it is that the overrides directory is always searched before its parent directory (the one it "overrides") whereas the local man tree /usr/local/share/man appears independently in the man tree search order.

4. Exit your shell to get your PATH and LANG back. Then start a new shell again. This time just delete your PATH variable using unset.

5. Examine your LANG variable and note the language prefix. (It is probably en). We will now set it to Spanish using LANG=es_ES.utf8. Then display your MANPATH again (remember, use /usr/bin/man --path). Did anything change when LANG is set? (The language-specific subdirectory /usr/share/man/es is searched before /usr/share/man, but it may not appear in the manpath.) Try /usr/bin/man nmap. Did the language change?

6. Set your PATH to /bin:/usr/bin. Don't forget export! Examine your MANPATH (using man --path) again. Did any major new directories get added? (Note: on RH7 an overrides directory may have been added. I believe this is a bug - it should have been added to the manpath when the PATH was empty.) However, no new major directories were added because the directives in /etc/man.config mapped each of your PATH directories to /usr/share/man. Add /usr/local/sbin (like this: PATH=/usr/local/sbin:$PATH) to the front of your PATH and see what happens to MANPATH. Can you explain it?

7. Next we will see what happens when you add an unfamiliar PATH directory. Add the directory /usr/java/latest/bin to your search path. Then examine MANPATH. What happened? By
exercising the /usr/java/latest directory, can you explain what happened?

8. To see how useful this is, examine the directory cs260a/bin beneath our public work area. In it you will find some commands. An example is the setmotd command. Try running setmotd using its absolute path. (It will fail.) Then try to find a man page for it. Next, add the directory it is in to your PATH and try to find the man page again. What happened? Can you explain this?

9. Exit your shell. Examine your original PATH and your MANPATH. Most of it should make sense now.

Exercise 5 - using man -k
If you don't know the name of the page you want, you must use man -k to search man's keyword database. On BSD systems man -k is named apropos.

man's keyword database (also called the whatis database) is created by the makewhatis command, which is often run periodically on linux overnight to keep the database up-to-date. makewhatis creates the database from the contents of the NAME section of each man page.

1. Unfortunately, if your topic is common, man -k topic will output too much information to be useful. Try it on directory. Note that it outputs the NAME section of each page it finds.

2. To narrow the search, the first solution is to limit the output to the type of page you want, as denoted by the section number. If you are searching for a command for directory, you could grep ‘(1)’ in the output. Try it.

3. If the results are still too broad you can add a second keyword by grepping again for another possible topic. Try appending a grep for ‘make’.

4. You might have assumed that the mkdir(1) command would be output by this search. Strangely, it is not. Examine mkdir(1)'s NAME section to see what the problem is.

5. Alter your previous command to eliminate the ending on the topic directory that indicates it is singular. Did mkdir(1) show up in the final output now?

Exercise 6 - other documentation sources
Although it is important to understand man pages and to be able to modify one if you need to, on linux manual pages are generated from a higher-level source format called texinfo. The intention of GNU/Linux (although they seem to have backpedaled a bit on this) was to replace the man system with info.

The info program reads a smaller list of larger topics and allows you to navigate interactively through all the info help on the system. The info database is organized into a ‘tree’ and you get individual help topics by ‘visiting’ nodes in the tree.

Try typing info on some topic that you want help on. To get used to the info system, get help on it (typing ? or cntl-h) and peruse the info commands. Don't be daunted; you can effectively navigate info using a simple set of commands:

- **move to the line describing the node you want to go to and press ENTER to go to it.**
- **SPACE** goes forward a page, BACKSPACE goes back a page.
- **n** goes to the next node. **p** goes to the previous node.
- **q** quits info

Other than that, investigate info on your own.

Two other URLs are useful for getting documentation on our systems:

http://docs.redhat.com

file:///usr/share/doc

Point a browser to each of this destinations and see the types of documentation they offer. Unfortunately, a lot of the product documentation in /usr/share/doc consists of stubs right now, but that is slowly changing. Some packages, such as pam, are surprisingly complete. And even the lowly README files often have a wealth of information. Check out a few. In the RedHat docs on their website, check out the manuals for RedHat Enterprise Linux, especially the Installation and Deployment Guides.
Answers and Hints

Exercise 3 - man trees
3. /usr/share/man/man1/javac.1.gz (It might not be gzipped)
4. at the time of this writing, there was a tree for Japanese. It is another man tree and should have directories like man1, etc. beneath it.

Exercise 4 - MANPATH
1. unset PATH LANG
5. the locale setting is la_CN.charset where la is the language, CN is the country and charset is the character set used. The only man tree that has an en subdirectory is /usr/share/man.
Even though you changed your language to Spanish, the manpath didnt change. This slight-of-hand is new in Redhat 7. It used to be explicit (by adding the language-specific subdirectory to the manpath) in RH6.
6. A directive in /etc/man_db.conf indicates that the PATH directory /usr/local/sbin should have its man pages in /usr/local/share/man. This is why the contents of the MANPATH did not change. You notice, however, that the order of the directories in MANPATH changed. This is because the man trees should be searched in the same order as the PATH directories. (The directive is called MANPATH_MAP in /etc/man_db.conf.)
7. Since /usr/java/latest had a man directory parallel to the bin directory you added to PATH, man found it when it created the default MANPATH.
8. Again, man added the man directory to MANPATH that was parallel to the new bin directory in PATH. (i.e., man looked for cs260a/man to correspond to cs260a/bin. Since it found the directory it added it to MANPATH.) It had the missing documentation.

Exercise 5 - using man -k
2. man -k directory | grep '(1)'
3. man -k directory | grep '(1)' | grep make
4. The problem is that the keywords (NAME section) of mkdir.1 contain directories, not directory.
5. man -k directory | grep '(1)' | grep make