

Chapter 1

1.1 For this exercise, we use the following two arguments for the `ls(1)` command: `-i` prints the i-node number of the file or directory (we say more about i-nodes in Section 4.14), and `-d` prints information about a directory instead of information on all the files in the directory.

Execute the following: `$ ls -ldi /etc/. /etc/.. -i` says print i-node number 162561 drwxr-xr-x 66 root 4096 Feb 503:59 /etc/. / 2drwxr-xr-x 19 root 4096 Jan 15 07:25 /etc/.. / `$ ls -ldi /. /..` both..and..have i-node number 2 2drwxr-xr-x 19 root 4096 Jan 15 07:25 ./ 2drwxr-xr-x 19 root 4096 Jan 15 07:25 ../

1.2 The UNIX System is a multiprogramming, or multitasking, system. Other processes were running at the time this program was run. 1.3 Since the `msg` argument to `perror` is a pointer, `perror` could modify the string that `msg` points to. The qualifier `const`, however, says that `perror` does not modify what the pointer points to. On the other hand, the error number argument to `strerror` is an integer, and since C passes all arguments by value, the `strerror` function couldn't modify this value even if it wanted to. (If the handling of function arguments in C is not clear, you should review Section 5.2 of Kernighan and Ritchie [1988].) 1.4 During the year 2038. We can solve the problem by making the `time_t` data type a 64-bit integer. If it is currently a 32-bit integer, applications will have to be recompiled to work properly. But the problem is worse. Some file systems and backup media store times in 32-bit integers. These would need to be updated as well, but we still need to be able to read the old format.

1.5 Approximately 248 days.

Chapter 1 Supplemental Exercises and Answers

1.6 EXERCISE: Most UNIX system programs are relatively small and designed to do one thing well. List at least three advantages to this approach.

SOLUTION: 1. Smaller programs are easier to maintain. 2. Smaller programs are easier to test. 3. It is easier to have confidence that a smaller program does what it is supposed to do than a larger program.

This text is associated with Stevens/Rago / Advanced Programming in the UNIX(R) Environment, Third Edition, ISBN-10: 0-321-63773-9 (ISBN-13: 978-0-321-63773-4). Copyright 2013 by Pearson Education, Inc. Do not redistribute.

Instructor Manual for Advanced Programming in the UNIX Environment 3

4. Small programs can be combined in interesting ways to solve new problems. For example, to find all of the misspelled words in a document, you might type `spell filename | sort | uniq`. This is superior to building a spelling checker into every document editor.

1.7 EXERCISE: Usually the CPU time of a program doesn't exceed the wall clock time (the running time) of the program. Explain the circumstances under which this can be false.

SOLUTION: On a multiprocessor, if an application is multithreaded, more than one processor can accumulate CPU time at the same time, so it is possible for the CPU time to exceed the running time of the program.