

COMP 530 — Introduction to Operating Systems

Fall 2008
Kevin Jeffay

Homework 1, August 22

Due: 2 PM, September 5

UNIX Programming Refresher

Write a C program on UNIX to read in lines of characters from standard input (*i.e.*, the keyboard) and write them as 80 character lines to standard output (*i.e.*, the “screen”) with the following changes:

- Every carriage return (newline) is replaced by a space, and
- Every adjacent pair of asterisks “**” is replaced by a “^”.

For this program, a “line of input” is defined as any sequence of zero or more printable and “whitespace” characters (tabs, spaces, and carriage returns). Each “80 character line” to be written to standard output is defined as 80 non-carriage return characters plus a carriage return (newline). Your program should process input lines until the end-of-file is reached at which point the program should terminate.

Your program should output *only* 80 character lines (with a carriage return after each line). For example, if your program has read in a line of input and is waiting for more data in order to make an 80-character line for output, and end-of-file is reached on standard input, then the program should terminate without printing out the partial output line. In addition, your program should not output any user prompts, debugging information, status messages, *etc.*

For the second requirement, pairs of asterisks should be replaced as they are seen. Thus, the string “abc***def” contains only one pair of asterisks and should be converted to the string “abc^*def”. The string “abc****def” contains two pairs of asterisks and should be converted to the string “abc^^def”.

The purpose of this assignment is to learn the basics of the C programming language and the use of UNIX program development tools. As stated here, the problem has nothing to do with operating systems. A later assignment will build upon this assignment and turn it into an operating systems problem.

Grading

For this (and most other programming) assignments you will “turn in” your program for grading by placing it in a special directory on a Department of Computer Science UNIX machine and sending mail to the TA. To ensure the TA can grade your assignments in an efficient and timely fashion, please follow the following guidelines *precisely*.

- Log on to a CS Department UNIX machine. (Any CS machine will work, however the machines *classroom.cs.unc.edu* (primary) and *swan.cs.unc.edu* (secondary) have been specifically dedicated for use by undergraduates.)
- In your UNIX home directory, create the directory structure: *comp530/submissions*. (That is, create the directory *comp530* in your home directory and inside this directory, create the directory *submissions*.)
- To allow the instructors and the TA to access the directory while restricting access for all others, execute the following commands:

```
fs sa ~/comp530/submissions system:anyuser none
fs sa ~/comp530/submissions jeffay read
fs sa ~/comp530/submissions <TASLogin> read
```

where <TASLogin> is the TA’s login id. All subsequent subdirectories you create should inherit these permissions, however, to be safe you should explicitly check that this is the case. If you enter the command:

```
fs la ~/comp530/submissions/x
```

(where *x* is the name of any subdirectory you might create) you should see something like:

```
Access list for /afs/cs.unc.edu/home/<userid>/comp530/submissions/x is
Normal rights:
cs-machines l
system:administrators rlidwka
<userid> rlidwk
jeffay rl
<TASLogin> rl
```

where <userid> is your login id. If you see a line for any other user or a line of the form “system:anyuser rl” then something is wrong and you should contact either the TA or the instructor immediately.

- For each assignment you will create a subdirectory with a name specified in the assignment. You must also name your program *exactly* as specified in the assignment. For this assignment you should name your final program “warmup.c” and store it in a directory named HW1 (inside your *~/comp530/submissions* directory).
- When you have completed your assignment you should put your program and any other necessary parts (header files, *etc.*) in the specified subdirectory and send mail to the TA (contact coordinates will be set out shortly via email), indicating that the program is ready for grading. *Do not change any of your files for this assignment after sending this mail!* If the timestamps on the files change you will be penalized for turning in a late assignment. If your program has a timestamp after 2:00PM on the due date it will be considered late. If you wish to keep fiddling with your program after you submit it, you should make a copy of your program and work on the copy and should not modify the original.

When you email the TA indicating that your assignment is ready for grading, please state your UNIX login in the email message. Since many of you have email addresses that are different from your UNIX login, we need to know your login in order to find your homework files.
- All programs will be tested on *classroom.cs.unc.edu*. All programs, unless otherwise specified, should be written to execute in the current working directory. Your correctness grade will be based solely on your program’s performance on *classroom.cs.unc.edu*. Make sure your programs work on *classroom!*
- The program should be neatly formatted (*i.e.*, easy to read) and well documented. In general, 75% of your grade for a program will be for correctness, 25% for “programming style” (appropriate use of language features, including variable/procedure/class names), and documentation (descriptions of functions, general comments, use of invariants, pre- and post conditions where appropriate).¹

¹ For this first assignment, correctness and programming style/documentation will each count for 50% of your grade.