Homework 2

Due Date: Monday, April 29, 2002, ***at 11:59PM*** **Points**: 100

UNIX System

1. (*10 points*) What program is running as process #1?

2. (*15 points*) Suppose I have a file called `x` in my current working directory. File `y` is a (hard) link to this file, and file `z` is a symbolic link to it also. I now give the command:

   ```
   mv x a
   ```

   What would the outputs of the following commands be, and why?

   a. cat x

   b. cat y

   c. cat z

C Programming

3. (*25 points*) Write a C program named words.c that reads a string from stdin as words, and prints each word and its line number on stdout. Loop until stdin's EOF, then terminate. A word is defined to be any contiguous sequence of alphanumeric characters. Use the fgets function to read the input a line at a time. Your program should handle lines of up to 100 characters. Don't bother to check for longer lines; you'll fix that in a later program.

   Your program should print one word per line. For example:

   Sample stdin

   ```
   Hello, there, my old friend!
   How are you today?
   I am very well, thank you!
   Goodbye ...
   ```

   Corresponding stdout

   ```
   1 Hello
   1 there
   1 my
   1 old
   1 friend
   2 How
   2 are
   2 you
   2 today
   3 I
   3 am
   3 very
   3 well
   3 thank
   3 you
   4 Goodbye
   ```

4. (*25 points*) The Fibonacci numbers play an important role in biology, mathematics, and other sciences. The first two numbers of the sequence are 0 and 1, and the numbers of the sequence are formed by adding the two previous numbers; so, the first few terms of the seuence are 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, …. Please write a program which takes an integer *n* as a command line argument, and prints the first `n` numbers of the sequence.

Debugging

5. (*25 points*) What does the following program do when you run it? Comment it, expanding each argument of the *printf* in your comment so that anyone can understand what each argument is in simplest form. Just make one header comment, and do not clean up the program!

```
main() { printf(&unix["\021%six\012\0"],(unix)["have"]+"fun"-0x60);}
```

Extra Credit

6. (*10 points*) If you wrote the Fibonnaci program in problem 4 using recursion, write a second version using a loop (and not using recursion). Conversely, if you wrote the Fibonnaci program using loops and not recursion, write a second version using recursion.