Trillion Counter

**Trillion Counter in C**

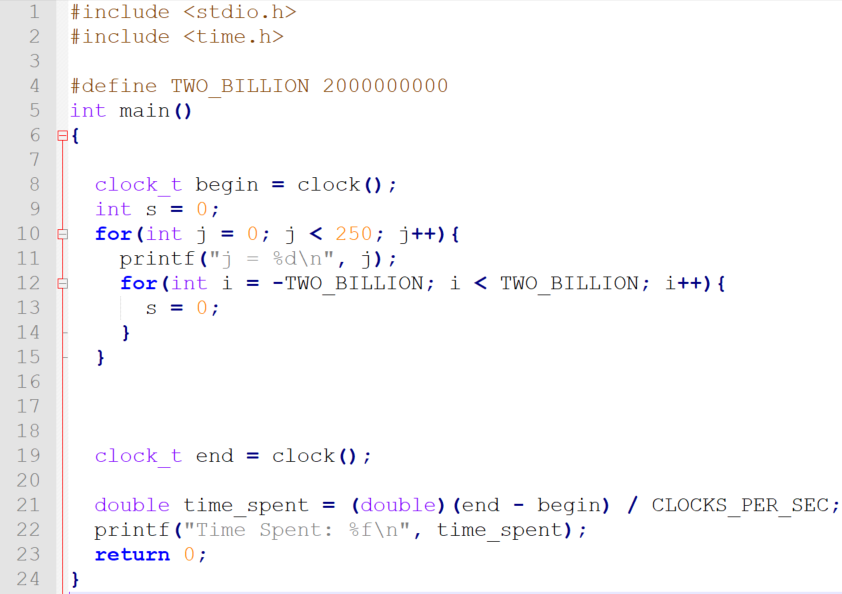


Figure 1 Trillion Counter C Program

The goal was to write a C program that has can be in 1 trillion distinct states. We call our program the “trillion counter” (Figure 1). Two int’s, i and j, where -2×109 ≤ i < 2×109 and 0 ≤ j < 250 were used to achieve the 1 trillion distinct states. The ranges of i and j give 4×109 distinct states for i and 250 distinct states for j. Nesting the incrementation of i and j gave 4×109×250 = 1012 or 1 trillion distinct states. First, the object file (flipped.o) was saved for bit manipulation. Only then the object file was linked and resulting program was run.

The execution time of the program was 2183.953125 seconds.

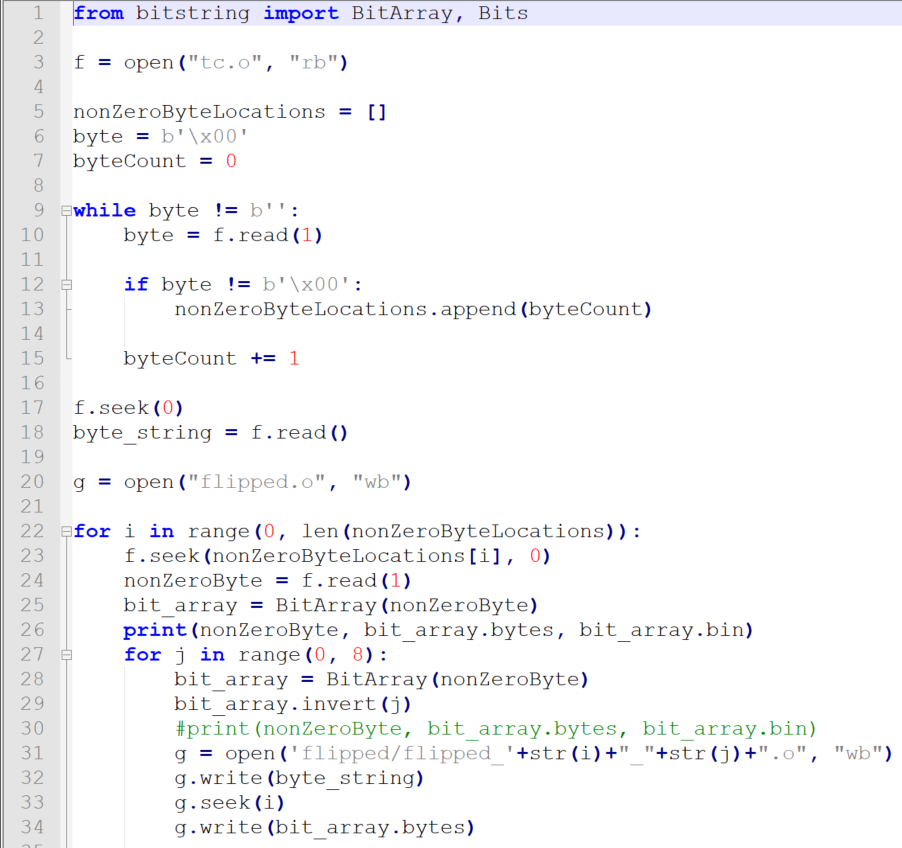
**Bit Flipping**

Figure Python script for flipping bits

Figure 2 shows the Python script for flipping bits that occur at non-zero bytes. After flipping each bit, the resulting binary was saved as object files. Since our program contained 4224 non-zero-byte bits, our python script created 4224 object files.

**Linking and Testing**

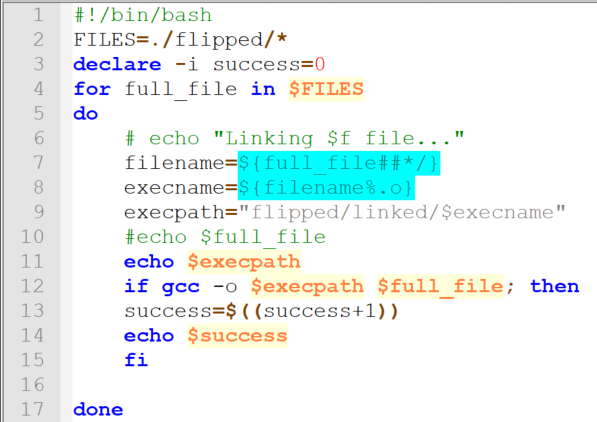


Figure 3 Bash script for linking object files

The next task was to attempt linking object files, and to calculate the percentage of object files that were successfully linked. The bash script showed that 3834 out of 4224 object files, or 90.77% linked successfully.

**Program Execution Testing**

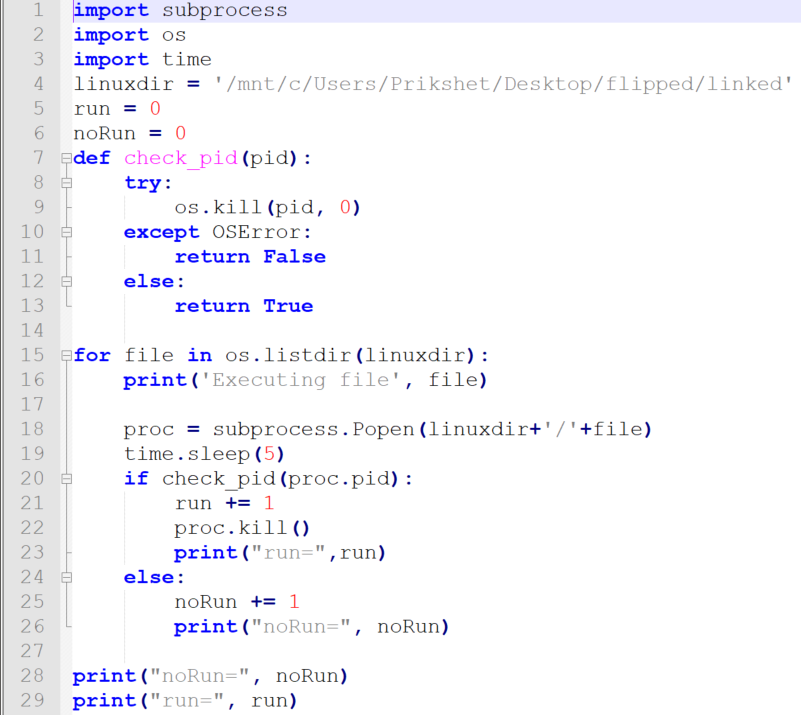


Figure 4 Python script for program check

The next task was to run the programs and count which the number of programs that run more than 5 seconds and the number of programs that run less than 5 seconds. The script stopped with “exec format error” after running 2837 program, out of which 222, or 7.82% terminated in less than 5 seconds.