1 Overview

Your assignment is to write a MIPS assembly language program that uses ShellSort to sort an array of integers stored in memory into ascending order.

2 ShellSort

Here is a Java version of ShellSort, plus a main program that tests it. This file is available at /home/cs340/prog2/sort.java. The very first thing you should do is convert this code into pseudo-code that uses only gotos and conditional gotos, as shown in class.

class sort {
    static void swap (int i, int j, int a[]) {
        int tmp = a[j]; a[j]=a[i]; a[i]=tmp;
    }
    static void shellsort (int a[]) {
        int n=a.length, incr=(n-2)/2;
        while (incr > 0) {
            for(int i=incr; i<n; i++) {
                int j = i-incr;
                while (j>=0) {
                    if (a[j]>a[j+incr]) {
                        swap(j, j+incr, a);
                        j=j-incr;
                    } else
                        break;
                }
                incr /= 2;
            }
        }
    }
    public static void main (String args[]) {
        int a[] = {12, 3, 76, -45, 99, 101, 11, -34};
        shellsort(a);
        for(int i=0; i<a.length; i++)
            System.out.print(a[i] + " ");
        System.out.println();
    }
}
3 Assignment

Write a MIPS assembly language program that implements a ShellSort routine that sorts an array of 4-byte signed (2’s complement) integers. The file /home/cs340/prog2/ssort.s contains a skeleton program to get you started. The integers array contains the integers to be sorted, and the count variable the number of them. Do not change their names. When your program finishes the values in integers should be sorted in ascending order. The file /home/cs340/prog2/Makefile contains a Makefile that you can use to turn in your program. I would make a subdirectory for this assignment, copy ‘ssort.s’ and Makefile into it, and work inside it.¹

It is not necessary for you to write separate procedures swap and shellsort as in the Java code above. If you want to you can simply copy the statements inside these routines to the place of their call. However, if you do implement swap and shellsort as separate procedures and use the correct MIPS calling convention you will get some bonus points. Note: swap can be implemented more easily in assembly code than in Java. Can you figure out how?

4 Honors Section

If you are in the honors section you should hand in two assembly code files: ssort.sp which runs under SPIM, and ssort.s which can be assembled, linked, and run on Wonka. Modify the Makefile so that ‘make ssort’ will produce a runnable program ssort on Wonka.

5 Turnin

When you have completed the program, submit your ssort.s file by typing ‘make turnin’. Or, to do it manually type ‘turnin cs340prog2 ssort.s’. You may turn in your ssort.s file as many times as you want; turnin will always replace the previously turned-in version with the new version. Leave count and integers set to the values in the skeleton ssort.s file. We will test other values during grading.

¹Use the Unix command ‘mkdir name’ to create a directory, ‘cd dir’ to change directories, and ‘pwd’ to see what directory you are in. ‘cd’ without any parameters takes you back to your home directory.