COMPSCI 210 S1T 2007 Tutorial Three ------- Assembly Code

Aim for the tutorial:

In this tutorial, we will learn how to run Alpha Simulator and familiar with the functionalities, also do some debugging exercises.

1. Introduction for using Alpha simulator:

Before we start use the Alpha simulator, we need download alpha, source code and setup Java Runtime Environment package.

1) Download alpha simulator and example source code

Unzip the exer.zip file, after then you will get the "AssemblyExercises" folder, you should get a directory structure like this:

AssemblyExercises

IMPORT
SYSTEM
example1
example2
example3

• Folder "IMPORT" and "SYSTEM" are two system directories. You should always keep them.

• Store you assembly files in folder like "example1". Two files you need, "*.user.s" and "Alpha.config".

• "*.user.s" stores your assembly code. The file name should be whatever the name followed by ".user.s".

• "Alpha.config" is a config file, which stores path of system files and your assembly file. Example of "Alpha.config" file:

../SYSTEM/palcode.pal.s
../SYSTEM/kernelcode.kernel.s
example1.user.s

What the Kernel and PAL code do?

 The Kernel code represents a very simple operating system. The only services this simple operating system provides are reading and writing characters, and terminating the user program.

 The PAL code can be thought of as implementing instructions that are too complex to be implemented in hardware.

We execute a call_pal instruction to execute a PAL code function. For example “call_pal CALL_PAL_CALLSYS” implements the “callsys” instruction to switch from executing user code to execute kernel (operating system) code. “call_pal CALL_PAL_RETSYS” implements the “retsys” instruction to switch from executing
kernel code to execute user code.

1.1 using alpha simulator run assembly code:

Step1: Run the alpha simulator
Right click on “simulator6.004.jar”, and then choose menu “open with”, then click on “java” or “javaw”, after execute the program we will see three window application in the screen.

• **Trace**. Display a trace of the recently executed instructions.
• **Registers**. Display the values of the registers (in our lecture we just using the integer registers).
• **User Memory**. Display the contents of user memory – the user program, global data, and function stack.

Step2: Load Files
To run a assembly program, firstly, you need to load the file “alpha.config”. To load the file, you can use menu “load file specification” in main menu “File”in any window. Or you can type “CTRL+N” in any window. Then select the file “alpha.config” which relate to your assembly code file. (“alpha.config” and code file will store in same folder).

Step3: Load Code
After you load up the file, then you need to Load Code. By choosing “Load Code” in menu “Run”, or type “CTRL+L”.

Step4: run assembly program
• “ctrl+X” (“Run/Rerun”): execute the program from the beginning, or
• “ctrl+R” (“Run/Continue”): run and stop in the breakpoint and resume execution from the point at which it stopped, or
• “ctrl+S” (“Step”): execute one instruction, and then stop.
After you execute run option we will get a new window application:
• **Simple Terminal**. Accept user input and display the output.

Option Step5: Set watch point for debug the code

To debug your assembly code you need to familiar with different ways to run the assembly program and how to set watch point You can set/clear watch points on a

range of selected memory or registers by selecting the menu item “Set Watch points” in the “Watch” menu or by typing “ctrl+w”. The simulator will stop the execution when the program attempts to perform the specified access.

Exercise:

1. Do an exercise to practice what we learned.
Objective: Load an assembly example “example1”, set several watch points, and watch changes of values of registers. You should get same value in register “a1” as your inputted char. For example, if you input char ‘a’, then you should get 61 in register a1.
Steps:

1) Set a watch point in the line “addq $zero, $v0, $a1”. After you load the code, click the window “user 0 memory”, then click the line showing the code, type “CTRL+W”
2) Run the program and stop in the first watch point by typing “CTRL+R”. A “Simple Terminal” will pop up firstly to ask you input a char. Suppose you input ‘a’ and press “Enter”, the execution should stop in the watch.
3) Find out which instruction is up to execute in next. Click on the window “Trace” and the last instruction in the bottom of the window. Here, it should be the instruction “addq $zero, $v0, $a1” because we have set a watch point to that line.
4) To watch the value of register “a1”, we want to only execute one more instruction, which is “addq $zero, $v0, $a1”, otherwise other instructions may change the value in “a1”. To do this by typing “CTRL+S”.
5) Read the value of register $a1 in the window “Register”

2. Debugging example.
This section will be showed in the tutorial!