COMP2611: Computer Organization

MIPS procedures

Overview

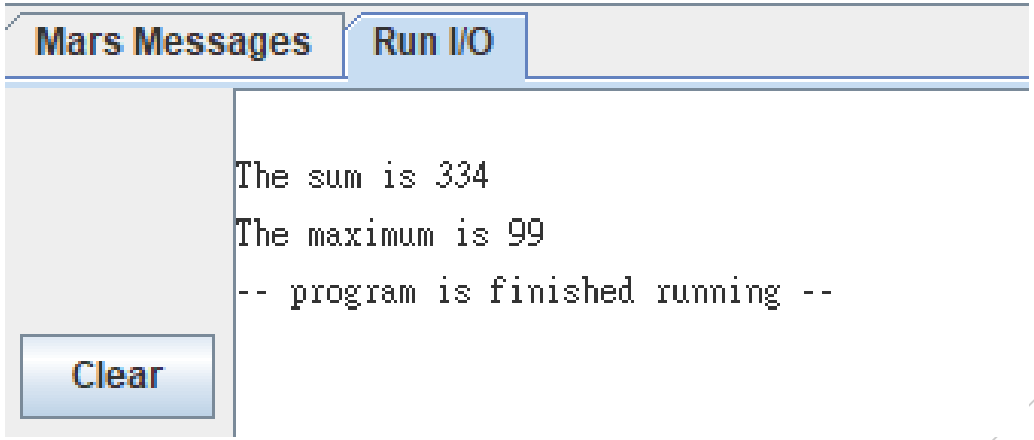
- You will learn the following in this lab:
  - How to use MIPS procedures in a program.
  - Store the values in the Stack (PUSH) and
  - Retrieve the stored values from the Stack (POP).

Example programs

- Try the following example program in order:
  - functionCall1.s (returning through the address in $ra).
  - functionCall2.s (can't return because $ra has been overwritten).
    - i.e. It cannot go to back to the "main" calling procedure and become infinite loop
    - Try to uncomment the lines following the "uncomment_solution", the $ra will be saved (pushed) to the stack before it is overwritten, and "Pop" out when it is needed.
  - functionCall3.s (preserving the registers).
    - $S is the preserved register. If those $S register is changed in any callee procedure, it needs to be restored back before returning to caller
  - functionCall4.s (passing more than 4 function arguments).
  - functionCall5.s (preserving the function arguments).

Exercise

- Please open the skeleton file in MARS: myArray.asm
- Complete the following tasks
  - Call a procedure findSum to get the sum of myArray
  - Call a procedure findMax to get the maximum value of myArray
  - You may need to preserve the value of any registers in case they will be overwritten

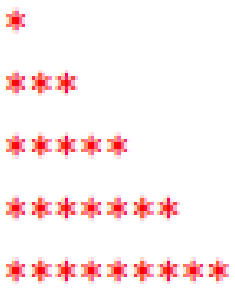
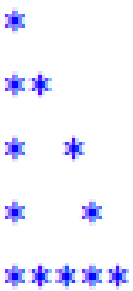


Challenge task: Can you also call a procedure to sort the array and output the result?

Extra Exercise

- The program starTriangle.s allow the users to set the size of RAT, and print it
- A RAT that has a size of 4 looks like this:

- Modify the program in order to call various procedures to output the following patterns.

Fat RAT	Hollow RAT	Upside-down RAT
		

Conclusion

- You have learnt:
  - How to use MIPS procedures in a program.