

Compiler Project – TD2 : SPIM

{Ghislain.Charrier, Bogdan.Pasca} @ens-lyon.fr
March 1st, 2010

SPIM

Please check out the SMIM, xspim, PCSpim documentation available at : http://perso.ens-lyon.fr/bogdan.pasca/teaching_projcompil.php.

A practical approach

Create a file with the following code :

```
.data
str: .ascii "abcd"

.text
.globl main
main: li $v0, 4
la $a0, str
syscall
```

Exercise 1 :

Run this code using the **step** mode so that you understand the global layout of a SPIM program. The output on the SPIM console will be : "abcd".

Exercise 2 :

Modify the above example so to print print the numbers : 42, 3.1415. Check out the different registers one has to use for printing different data types using the **syscall**.

Exercise 3 :

Using SPIM, write and test an adding machine program that repeatedly reads in integers and adds them into a running sum. The program should stop when it gets an input that is 0, printing out the sum at that point. Use the SPIM system calls.

Exercise 4 :

Using SPIM, write and test a program that reads in three integers and prints out the sum of the largest two of the three.

Exercise 5 :

Using SPIM, write and test a program that reads in a positive integer using the SPIM system calls. If the integer is not positive, the program should terminate with the message "Invalid Entry"; otherwise the program should print out the names of the digits of the integers, delimited by exactly one space. For example, if the user entered "728," the output would be "Seven Two Eight."

Exercise 6 :

Write and test a MIPS assembly language program to compute and print the first 100 prime numbers. A number n is prime if no numbers except 1 and n divide it evenly. You should implement two routines :

- test_prime (n) : Return 1 if n is prime and 0 if n is not prime.
- main() : Iterate over the integers, testing if each is prime. Print the first 100 numbers that are prime.

Test your program by running them on SPIM.