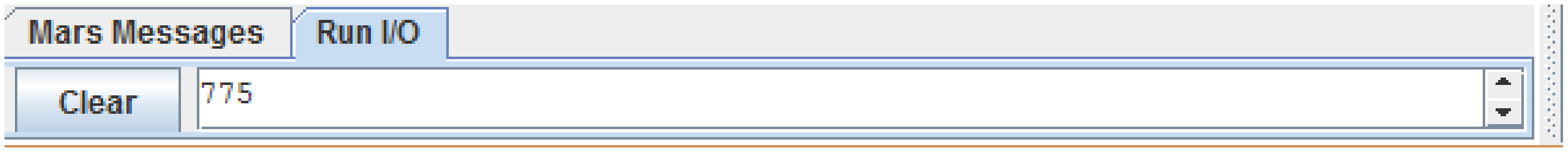
Directives

- Chỉ thị assembler lưu hoặc cấp phát program objects.
- Program structure
  - .text: store objects in the code segment (these are instructions)
  - .data: store objects in data segment (static variables)
  - .ktext, .kdata: code and data segments for kernel (OS)
- Variable declaration (allocation)
  - .byte/.half/.word: store listed values as bytes/halfs/words
  - .ascii/.asciiz: string and null-terminated string
  - .space: reserved specified number of bytes

syscall

- **Print decimal** integer to standard output (the console).
- Argument(s):
  - $v0 = 1
  - $a0 = number to be printed
- Return value: none

```
li   $v0, 1               # service 1 is print integer
li   $a0, 0x307           # the interger to be printed is 0x307
syscall                   # execute
```

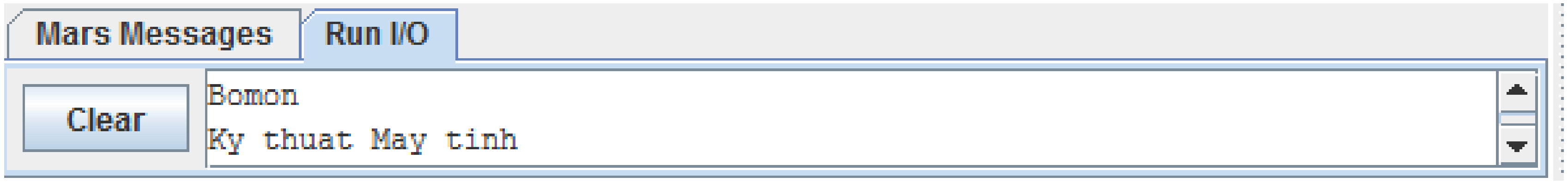


```
775
```

syscall

- **Print string** to standard output (the console).
- Argument(s)
  - $v0 = 4
  - $a0 = address of null terminated string to print
- Return value: none

```
.data
Message: .asciiz "Bomon \nKy thuat May tinh"
.text
   li  $v0, 4
   la  $a0, Message
   syscall
```



```
Bomon
Ky thuat May tinh
```

Hello World

```
.data   #Data segment
x:          .word    0x01020304    # x is a word
message:   .asciiz  "Dept. of Computer Engineering"
.text   #Code segment to store instructions
   la   $a0, message          #load string address to a0
   li   $v0, 4                #function $v0 = 4
   syscall                    #call system routine

   addi $t1,$zero,2           #$t1 = 2
   addi $t2,$zero,3           #$t2 = 3
   add  $t0, $t1, $t2         #$t0 = $t1 + $t2
```

- Chạy và quan sát
  - Text, data segment
  - la, li, syscall
- Bài tập: viết các lệnh để gán giá trị 0x2023 cho x

syscall

- **Read integer** from standard input (the console).
- Argument
  - $v0 = 5
- Return value
  - $v0 = contains integer read

```
li      $v0, 5
syscall
```

syscall

- **Read string** from standard input
- Argument(s):
  - $v0 = 8
  - $a0 = address of input buffer
  - $a1 = maximum number of characters to read
- Return value: none
- Note: for specified length n, string can be no longer than n-1.
  - If less than that, adds newline to end.
  - In either case, then pads with null byte
- String can be declared with *.space*

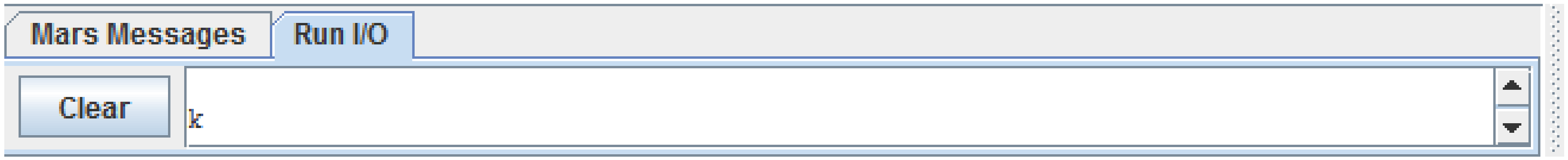
syscall

```
.data
Message: .space 100     # string with max len = 99
.text
   li  $v0, 8
   la  $a0, Message
   li  $a1, 100
   syscall
```

syscall

- **Print** a **character** to standard output.
- Arguments
  - $v0 = 11
  - $a0 = character to print (at LSB)
- Return value: none

```
li   $v0, 11
li   $a0, 'k'
syscall
```



```
k
```

syscall

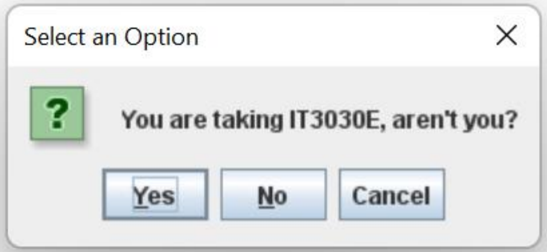
- **Read** a **character** from standard input.
- Argument(s):
  - $v0 = 12
- Return value:
  - $v0 contains the character read

syscall

- ConfirmDialog
- Argument(s):
  - $v0 = 50
  - $a0 = address of the null-terminated message string
- Return value: $a0 = value of selected option

  0: Yes 1: No 2: Cancel

```
.data
Message: .asciiz “You are taking IT3030E, aren’t you?"
.text
    li    $v0, 50
    la    $a0, Message
   syscall
```


Select an Option

You are taking IT3030E, aren't you?

syscall

- exit: terminate the program
- Argument
  - $v0 = 10
- Return value: none