Hello World Assembly Example:

Objective

1. The aim of this activity is to give students know the system call in Ubuntu (Linux OS).

2. The second is to give the experience writing a code with assembly language to call the system call.

What is System Call?

System call is a function provide by OS for the application request services. Linux in kernel 2.0 has number of the system call at 190 functions in **32bit OS**, from table 5.1 shown only first 15 system calls. This activity uses system call no. 1 and 4.

No / System call	Short description
0x00	sys_setup Call function filesystem.c
0x01	sys_exit Terminate the current process
0x02	sys_fork Create a child process
0x03	sys_read Read from a file descriptor
0x04	sys_write Write to a file descriptor
0x05	sys_open Open if possibly create a file or device
0x06	sys_close Close a file descriptor
0x07	sys_waitpid Wait for process termination
0x08	sys_creat Create a file or device
0x09	sys_link Make a new name for a file
0x0a	sys_unlink Delete a name and possibly the file it refers to
0x0b	sys_execve Execute program
0x0c	sys_chdir Change working directory
0x0d	sys_time Get time in seconds
0x0e	sys_mknod Create a directory or special or ordinary file
0x0f	sys_chmod Change permissions for a file

Activity:

1. After installing nasm:
   $ sudu apt install nasm
2. Type the following ***hello.asm*** code into your editor *($ nano hello.asm)* :
3. Using the proper x86 commands assemble, link and run your program:
   $ nasm –f elf hello.asm
   $ ld –o hello_exe –m elf_i386 –s hello.o
   $ ./hello_exe

```
section .text
global _start
_start:
        mov edx,len                     ;message length
        mov ecx,msg                     ;message to write
        mov ebx,1                       ;file descriptor (stdout)
        mov eax,4                       ;system call number (sys_write)
        int 0x80                        ;call kernel
        mov eax,1                       ;system call number (sys_exit)
        int 0x80                        ;call kernel
section .data
        msg db 'Hello, world!',0xa      ;our dear string
        len equ $ - msg                 ;length of our dear string
```