Arquitecturas Intel 64 e IA-32

Organización del Computador II

24 de marzo de 2022

Actividad Individual

La presente guía corresponde a la clase de Introducción a Assembler Intel 64 e IA-32. Se trata de una primera aproximación al tema.

A continuación, les solicitamos completar una serie de ejercicios requeridos para su formación.

Ejercicio 1 Creación de la cuenta individual de GIT de la materia

Cada estudiante deberá tener una cuenta del Departamento de Computación que le dará acceso a un cuenta de email y al Git del DC.

Si ya poseen la cuenta, no es necesario crear otra. Verifiquen que puedan acceder al gitlab: `https://git.exactas.uba.ar/users/sign_in`

Ejercicio 2 Instalación de Software requerido para la materia

Les pedimos que se armen un entorno de trabajo con el software que precisarán para la materia.
Pueden encontrar algunos pasos orientativos en el campus bajo la solapa "Material de Cursada".

a) Linux

b) GCC

c) NASM

d) Bochs

e) Valgrind

f) Git

g) Editor de texto de preferencia (por ejemplo, sublime, vim u otro)

Ejercicio 3 Hola mundo

a) Escriba el programa Assembler "Hola Mundo"presentado en la clase práctica. Compilelo y pruebelo en sus computadora acorde a las indicaciones provistas en clase.

b) Modifique el programa para que imprima su nombre, apellido, número de libreta y alguna frase que les guste o motive

c) Pruebe el programa y suba el código de este último programa al Git. Recuerde que puede tener un .gitignore para subir únicamente el código del programa y no archivos objetos o ejecutables.

Ejercicio 4 Calculadora pobre

Esta es una actividad exploratoria en la cual buscamos se animen a ver que pasa si toman diferentes decisiones y valores en su código. Hay muchas soluciones, anímense a experimentar. Pueden usar el manual o consultar internet.
Van a necesitar usar **GDB**

GDB es un debugger de código, les permite ir ejecutando el código línea por línea e ir inspeccionado las variables, valores de los registros, estado de la memoria, etc.

Si siguieron el instructivo deberían ternerlo instalado (esta en el paquete build_essentials de Linux)

Para iniciar el *gdb* escriban en la terminal: gdb *nombre_archivo_ejecutable*

Dentro de gdb tienen un prompt.

- Para poner un breakpoint en la primer linea pongan: **b 1**
- Con **r** inician la ejecución
- Con **n** avanzan a la siguiente línea
- Con **c** continua la ejecución
- Con **info reg *nombre_registro*** pueden inspeccionar los valores de los registros. Por ejemplo, **info reg rax** , otro ejemplo: **info reg eflags**
- Con **q** pueden salir de gdb

a) Escriba un programa que sume dos números de 8 bits. Recuerden agregar la llamada al sistema operativo para indicar que termina el programa con la syscall sys_exit.

b) Investigue que pasaría si el resultado de la suma supera los 8 bits usando **GDB**.

c) Ahora, escriba un programa que sume dos números de 64 bits.

d) Pruebe su código sumando combinaciones de positivos, negativos y ceros.

e) Busque qué tipo de representación usa Intel. ¿En qué casos se activarían los flags de Zero, Carry y Overflow? Arme algún ejemplo numérico con el que logre activar estos flags. Compare el caso 8 bits y 64 bits. Inspeccione con gdb el registro EFLAGS y los registros que utiliza en la suma.

f) Punto Extra - Agregue a su programa una condición que imprima en pantalla el resultado si no hubo overflow u overflow si lo hubo. Recuerde verificar los flags.

g) Recuerde guardar su trabajo en el Git

h) Compare su solución con las soluciones individuales de su grupo y discutan las decisiones que fue tomando cada uno y los descubrimientos hechos.