

# Universidad Nacional Autónoma de México Facultad de Ingeniería



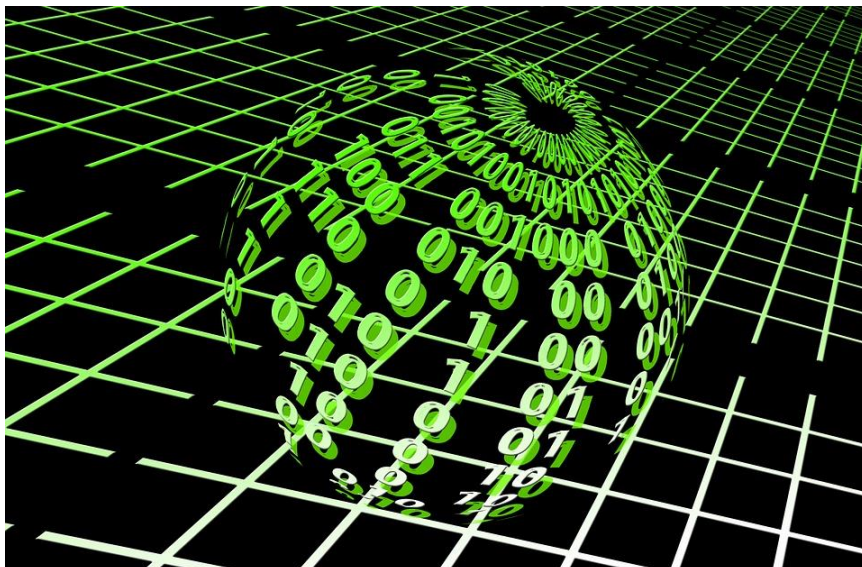
## División de Ingeniería Eléctrica Sistemas Operativos “Cargado y Ligado”

Expositores:

Martínez Garduño Héctor Alejandro

Materia: Sistemas Operativos

Grupo: 6



**Objetivo:**

Exponer sobre un tema a mi elección, el cual se tuviera en conocimiento paralelo, de la materia, ya sea dentro del temario o externo.

**¿Porque la decisión del tema?**

Para conocer el funcionamiento de un compilador así como se maneja un lenguaje desarrollado por un programador, y el cómo lo interpreta una computadora en cuestión.

**Exposición:**

Cargado y ligado de programas

Para que un programa funcione, se necesita :

- un compilador que realiza la función de cargar un programa
- un enlazador (ligado) este liga dicho programa.

¿Para que?

El trabajo básico de cualquier enlazador y cargador es simple: vincula nombres abstractos a nombres concretos, lo que permite a los programadores puedan escribir código utilizando los nombres más abstractos.

Compilación (Traductor o Cargado)

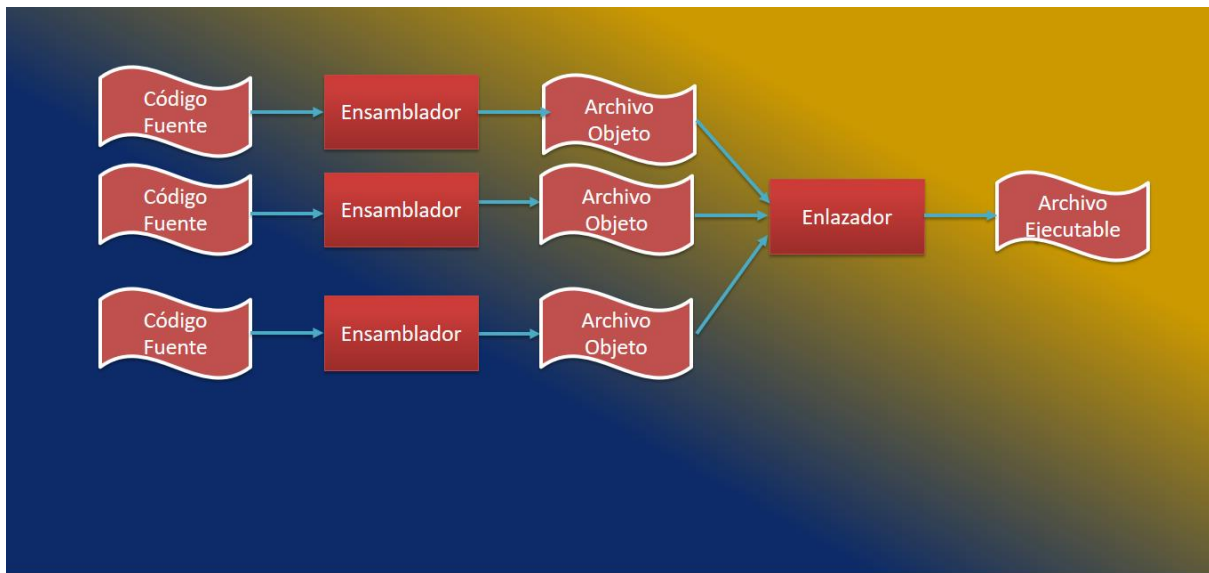
Un cargador es un procesador de lenguajes que toma programas en un cierto lenguaje fuente y los transforma en programas funcionalmente equivalentes, codificados en otro lenguaje tipo objeto.

Cargador de programas

Parte del sistema operativo que trae un archivo ejecutable que residen en el disco en memoria y empieza a correr

Los pasos que sigue para cargarlo

- Leer el encabezado del archivo ejecutable para poder determinar el tamaño de los segmentos de texto y datos
- Crear un nuevo espacio de dirección para el programa
- Copia las instrucciones y datos en espacio de direcciones
- Copia los argumentos y son pasados a la pila
- Inicializa los registros de la máquina incluyendo la pila
- Salta a una rutina de inicio que copia en el programa, con los argumentos de la pila como las llamadas y los registros de la rutina principal del programa



- Ensamblador (enlazador)

-Convierte los programas en lenguaje ensamblador dentro de un archivo objeto

- Los archivo objeto contienen una combinación de instrucciones de máquina, así como también datos e información que se necesitan para la memoria

Tres tareas del enlazador

- Buscar en el programa, rutinas de biblioteca utilizadas en el programa, por ejemplo. printf (),scanf()),for´s, rutinas matemáticas, etc.
- Determina las ubicaciones de memoria que codifican cada módulo que ocupará y reubicará sus instrucciones para ajustar las referencias absolutas
- Resuelve referencias entre archivos

Estructura de un archivo objeto



Encabezado de archivo de objeto describe el tamaño y la posición de la otra pieza del archivo

Segmento de texto contiene las instrucciones de la máquina

Segmento de datos contiene la representación binaria de los datos en archivo de ensamblado

Información de Reubicación: identifica las instrucciones y los datos que dependen las direcciones absolutas

Tabla de símbolos: asocia direcciones con etiquetas externas y listas de referencias.

Los Enlazadores necesitan traducir instrucciones de montaje y pseudo-instrucciones en las instrucciones de la máquina

– Convertir números decimales a programa en forma binaria.

Normalmente los montadores hacen dos pasadas el archivo ensamblado:

-Primer paso: Lee cada línea y graba etiquetas en una tabla de símbolos

-Segundo paso: utilizar información de la tabla de símbolos para producir código de máquina para cada línea

### **Conclusión:**

Como tal el tema es confuso, y muy extenso, por lo que me di a la tarea de conocer a grandes rasgos sin necesidad de entrar a las trivialidades. Se expuso lo mejor, y lo más entendible posible, ya que muchos no se entienden en los conceptos que se usan usualmente.

### **Bibliografía:**

- ☞ Desconocido. (). Assembler. Consultado el 21 de Octubre de 2017. Recuperado de <https://cs.gmu.edu/~setia/cs365-S02/assembler.pdf>, Traducción Hector Martinez
- ☞ Kaufman Morgan, (), Linkers and loaders, Why do we need linkers and loaders, Consultado el 21 de octubre de 2017, Recuperado de <https://www.iecc.com/linker/>, Traducción Hector Martinez
- ☞ Tanenbaum Andrew, (2003), Sistemas Operativos Modernos, Edición tercera, Consultado el 19 de octubre de 2017
- ☞ Milenkovic Milian, (1996), Sistemas Operativos: concepto y diseño, Edición Segunda, Consultado el 19 de octubre de 2017

- 3 Espinoza A. Eduardo ()Principios de diseño de lenguajes, Consultado 15 de octubre de 2017, Recuperado de <http://dicyg.fi-c.unam.mx:8080/lalo/lp/presentaciones/principios-de-diseno-de-lenguajes>
- 3 Espinoza A. Eduardo ()Tipos de datos, Consultado 15 de octubre de 2017, Recuperado de <http://dicyg.fi-c.unam.mx:8080/lalo/lp/presentaciones/tipos-de-datos>