



Sistemas Operativos Avanzados

Apuntes: 6 Febrero 2017

David Rojas Leitón

Lectura de la carta del estudiante

Datos importantes del curso:

Email de soporte: soporte@ic-itcr.ac.cr

Dirección del foro: www.ic-itcr.ac.cr/foro

Email del profesor: torresrojas.cursos@gmail.com

Carta del estudiante:

- Apuntador y Apuntes
 - En cada clase habrá al menos 2 apuntdores
 - Hay 4 días hábiles para subir los apuntes al foro
 - El formato del reporte debe ser PDF
 - El formato para el nombre de los apuntes es el siguiente: **SOA17mmdd-k.pdf**, donde *mm* es el mes, *dd* la fecha del día y *k* un número secuencial para distinguir entre todos los apuntes de la misma clase
 - Los apuntes son una cosa seria, todos eventualmente lo van a hacer es obligatorio, mínimo un par de veces, los apuntes tienen puntos en la nota y deben ser subidos antes de la clase
- Ausencias
 - La presencia en clase no es obligatoria, pero lo recomendable es ir para poder tener voz y voto en decisiones relacionadas al curso
 - Si falta no le escriba por nada del mundo al profesor pidiendo la materia
- Exámenes
 - Todo el material será evaluado en 2 exámenes parciales no acumulativos duro el semestre
- Resúmenes
 - La idea de leer papers muy viejos es conocer cosas que aún no han cambiado o ver como ha sido la evolución de otras
 - De los *N* papers estudiados durante el semestre, cada estudiante debe entregar *K* resúmenes escritos, donde *K* es el mínimo entre 25 y 60% de *N*
 - Bajo ninguna circunstancia se aceptará la entrega atrasada de resúmenes
 - Si se entrega más de un resumen por clase cada uno debe venir de forma individual
 - En la primera parte un resumen del material del paper
 - En la segunda parte se deben contestar las siguientes preguntas:
 - i. ¿Cuál es el problema que plantea el paper?
 - ii. ¿Por qué es un problema interesante o importante?
 - iii. ¿Qué otras soluciones se han intentado para resolver este problema?
 - iv. ¿Cuál es la solución propuesta por los autores?
 - v. ¿Qué tan exitosa es esta sola solución?

- Presentaciones orales
 - Todos los estudiantes del curso deben presentar durante el semestre un paper asignado por el profesor
- Papers de investigación
 - Cada grupo de trabajo de los proyectos programados debe escribir un artículo científico de temas relacionados a Sistemas Operativos Avanzados
- Proyectos Proyectos
 - Se harán 3 proyectos
 - En grupos de 4 o 5 personas
- Tareas y quices
 - Al final del semestre se tomarán en cuenta únicamente los 8 mejores quices de cada persona

Formato que debe tener cada resumen

<div style="border: 1px solid black; padding: 10px; margin-bottom: 10px;"> Datos del estudiante y resumen </div>		<div style="border: 1px solid black; padding: 10px; margin-bottom: 10px;"> Código del paper </div>
...		

Aclaraciones del profesor:

- El curso es muy entretenido, importante y lleno de conocimiento
- El profesor tiene la mayor voluntad de que los estudiantes aprendan
- Hay que leer mucho
- Las evaluaciones del curso son formales y rigurosas
- Es importante dedicarle “ratillos” al curso para tener éxito
- No hacer fraude. En caso de comprobarse un fraude será severamente castigado
- La calidad académica es la principal prioridad de la maestría
- El profesor no quiere dejar a nadie
- El curso se gana con 67.5 que se redondea a 70
- No se dan 80s para ayudar a alguien a mantener promedios
- El curso es presencial
- Si se pueden instalar equipos para acceso remoto, pero sin involucrar al profesor ni ser usado como excusa en nada
- Las reposiciones de evaluaciones serán donde, cuando y a la hora establecida por el profesor
- La libertad de cátedra del profesor se respeta
- El horario y cronograma solo será alterado por cosas de fuerza mayor

- La puntualidad se respeta
- Las actividades sociales o laborales no son excusa para faltar con evaluaciones o lecciones
- Si un estudiante sabe de antemano que va a faltar por muchas semanas al curso por asuntos de trabajo no deberá matricular el curso
- Canal de comunicación primario = el profesor, debe de agotarse este medio antes de ir a hablar con alguien más sobre alguna molestia
- Si se dan comunicaciones incorrectas serán rechazadas y devueltas al profesor



Nota importante: Se consultó sobre la evaluación y todos los estudiantes presentes estuvieron de acuerdo.

Repaso de conceptos básicos de sistemas operativos

Nacimiento de los lenguajes de bajo nivel



John William Mauchly

- Físico U.S.A
- Ph. D. Física - John Hopkins University
- Se interesó por electrónica en 1941



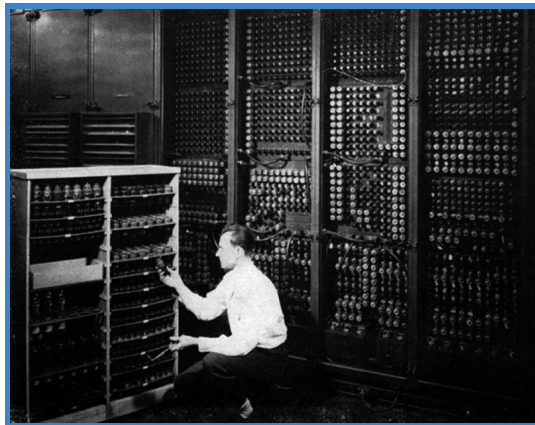
John Adam Presper Eckert



- Ingeniero Eléctrico, U.S.A (1919 - 1995)
- Conoce a Mauchly en 1941

ENIAC (Electronic Numerical Integrator And Computer)

- Primer computadora electrónica programable
- Para programarla había que cambiar los cables de lugar
- Aunque fuera tan rudimentaria se dice que es programable porque es una máquina capaz de hacer varias cosas
- Fue llamada como Giant Brain por la prensa para su publicidad
- Las mujeres fueron las primeras programadoras, la programación se consideraba femenina basado en eso
- Resolvía problemas de balística, pero se podía cambiar el programa
- Funcionaba solo la mitad del tiempo



EDVAC (Electronic Discrete Variable Automatic Computer)

- Se empezó a trabajar en ella antes que ENIAC
- Funcionaba 20 horas diarias
- MTBF (Mean Time Between Failures): 8 horas
- Almacena programas en memoria, creando con esto el software
- John von Neumann fue consultor de su construcción



ENIAC - EDVAC

	Creada por	Era “programmable”	Aguantaba programas en memoria	Horas de trabajo	MTBF
ENIAC	Universidad de Pensilvania	Si	No	8	—
EDVAC	Universidad de Pensilvania	Si	Si	20	8 horas

John von Neumann

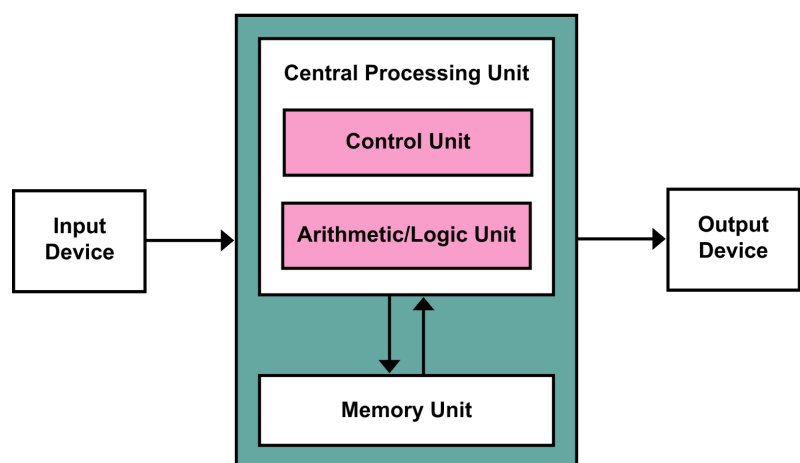
- Considerado uno de los mejores matemáticos del siglo XX
- Uno de los científicos más brillantes de la historia
- Fue parte del proyecto Manhattan
- Sus trabajos fueron desde arquitectura de computadoras, ciencia de la computación, investigación de operaciones, teoría de juegos, teoría de conjuntos, economía, física, etc.
- Fue consultor de la construcción de EDVAC



Arquitectura de von Neumann

Esta arquitectura fue propuesta por von Neumann en 1945, pero se dice que las ideas presentadas en el paper son robadas de una reunión a la cual asistió, también de algunas ideas de las correcciones que le hizo Alan Turing a su trabajo. Este paper es conocido como el First Draft.

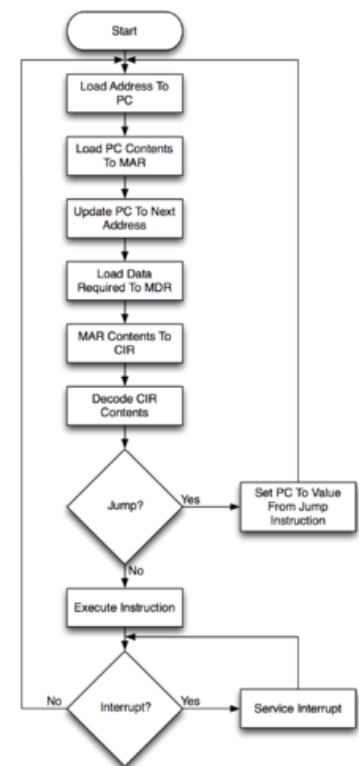
La arquitectura describe un computador con un CPU, una memoria, entradas y salidas. Gracias a esta arquitectura apareció el ciclo de fetch, programas que se leen de una memoria, el concepto de conjunto de instrucciones y el concepto de Software.



Unidad de Control

La unidad de control es la encargada de ejecutar el ciclo de fetch, la cual sigue los siguientes pasos para ejecutar la tarea:

1. Recupera de memoria la siguiente instrucción (PC)
2. Actualiza PC
3. Decodifica una instrucción
4. Decodifica los argumentos (modos de direccionamiento)
5. Ejecuta la instrucción
6. Revisa las interrupciones
7. Retornar al paso #1



Diseño de la Unidad de Control

La unidad de control está ligada al lenguaje máquina o la arquitectura de una computadora y tiene dos posibilidades básicas de diseño, una de ellas es la alambrada y la otra es la microprogramada.

1- Unidad de Control Alambrada

En la unidad de control alambrada se implementan las puertas lógicas y cada instrucción de lenguaje máquina tiene sus circuitos correspondientes. También se debe tener presente que es una arquitectura fija, lo que significa que no se puede cambiar, en caso de querer hacerlo es necesario rediseñar la unidad de control lo que las convierte en poco o nada accesible. Estas unidades son mucho más rápidas y eficientes que las unidades de control microprogramadas.

Sir Maurice Vincent Wilkes

- Físico inglés
- Fue uno de los fundadores de la ciencia de la computación y la investigación de las operaciones
-
- Fue el inventor del concepto de **microprogramación** en 1951 para el diseño de unidades de control en CPUs



- Contribuyó con el concepto de **subrutinas en FORTRAN**
- Inventó **EDSAC (Electronic Delay Storage Automatic Calculator)** que fue una computadora electrónica
- Ganó el Turing Award en 1967

2- Unidad de Control Microprogramada

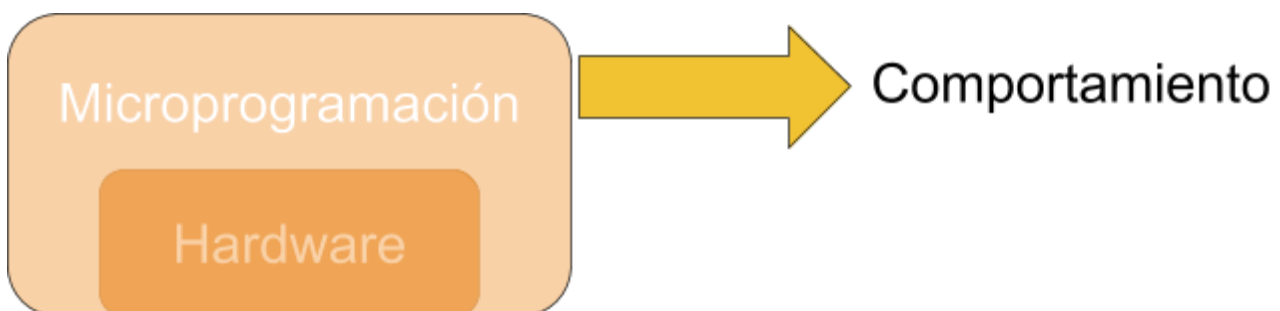
En una unidad de control microprogramada existe una organización subyacente, desde donde se crea la arquitectura y el conjunto de instrucciones ejecutadas se define sobre esa organización, cada instrucción de lenguaje máquina se construye con múltiples microinstrucciones, que son tomadas de una micromemoria. Estas unidades de control son más flexibles, pero también más lentas.

Microprogramada \neq Microprogramable

Microprogramada: No se puede cambiar

Microprogramable: Si se puede cambiar

- Lo único real del real de la unidad de control microprogramable es el hardware
- Esta unidad se rodea microprogramación para darle un comportamiento
- Teóricamente con una unidad de control microprogramada podríamos hacer lo que sea, pero el tiempo que tomaría hacerlo sería mucho



Lenguaje Máquina

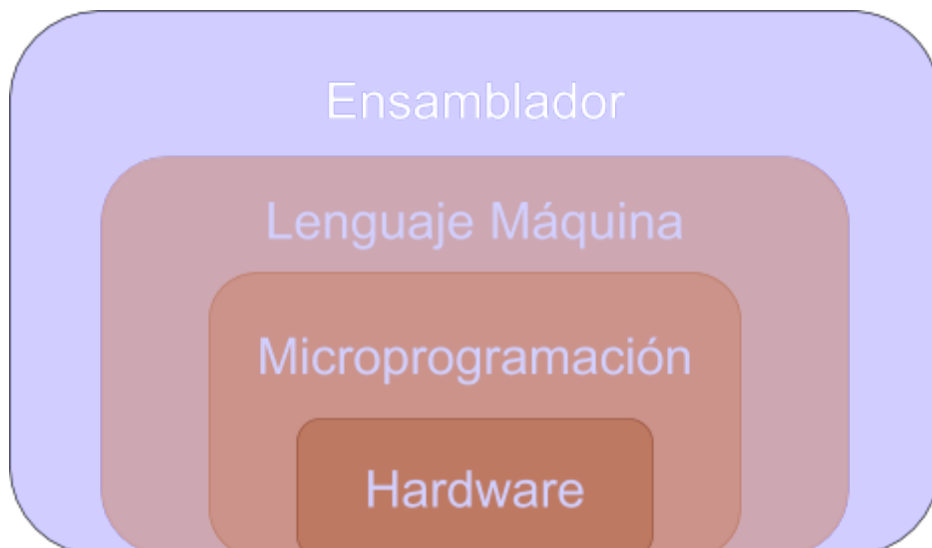
El lenguaje máquina es definido por la microprogramación. Fue creado para simplificar el trabajo de crear software con la microprogramación. Este lenguaje no es lo mismo que ensamblador.



No todo lo que yo puedo hacer en microprogramación lo puedo hacer en lenguaje máquina, pero todo lo que puedo hacer en lenguaje máquina lo puedo hacer en microprogramación.

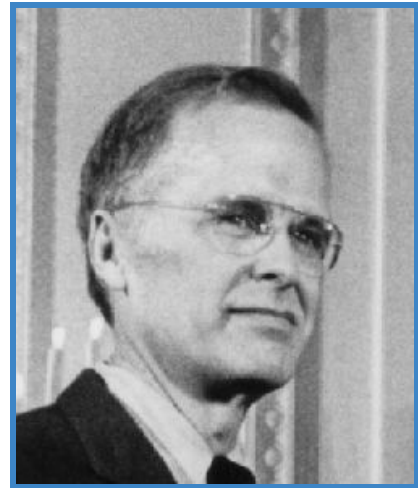
Lenguaje Ensamblador

El lenguaje ensamblador fue inventado en los años 50 y viene a mejorar la experiencia tediosa y propensa a errores del lenguaje máquina. Básicamente es un programa que recibe nemónicos y nombres simbólicos con cierta sintaxis simple y genera lenguaje máquina. El primer ensamblador fue escrito en lenguaje máquina.



John Backus


- Matematico y científico de la computación
- M. Sc. en Matemáticas - Columbia University
- Trabajo en IBM desde 1950 programando lenguaje máquina y ensamblador
- Su primer proyecto fue un programa para calcular las posiciones de la Luna
- Propone en 1953 el proyecto FORTRAN (Formula Translation) para la IBM 704
- Inventó el Speedcoding para hacer más fácil programar




Los 4 grandes


Entre 1950 y 1960 existieron 4 grandes lenguajes de programación, ellos son:

 **FORTRAN** - Creado por John Backus - Ganador del premio Turing en 1977

 **LISP** - Creado por John McCarthy - Ganador del premio Turing en 1971

 **ALGOL** - Creado por Peter Naur - Ganador del premio Turing en 2005

 **COBOL** - Creado por Grace Hopper* - Nunca ganó el premio Turing

*  **Grace Hopper:** Nació en Estados Unidos donde no solo fue experta en computación sino que también fue contralmirante del ejército. **Fue una de los primeros programadores de la computadora Harvard Mark I e inventó el primer compilador para un lenguaje de computación.** Ella **popularizó la idea de los lenguajes de programación de máquina-independiente**, lo que llevó al desarrollo de COBOL. También **popularizó el término bug**.

Lenguajes de Alto Nivel

Los lenguajes de alto nivel existen desde 1956 y han alcanzado fama debido a la facilidad que estos representa a la hora de programar, en otras palabras la productividad creció mucho con ellos. Al igual que los otros lenguajes estos definen una nueva capa con independencia del hardware y una mayor portabilidad.



Productividad vs Eficiencia

Productividad: Líneas de código sin errores por unidad de tiempo al desarrollar un sistema

Eficiencia: Cantidad de recursos (tiempo, espacio, etc.) que un sistema en funcionamiento requiere por cada resultado



La Eficiencia no sufre entre Lenguaje Máquina y Ensamblador porque todo lo que puedo hacer en un lenguaje también lo puedo hacer en el otro.

Sistemas Operativos Distribuidos

IBM 701

Fue la primer computadora comercial y solo se fabricaron 19, al ser tan pocas no se vendió ninguna solo se alquilaban. Aceptaba 150 tarjetas por segundo de entrada y 100 tarjetas por segundo de salida. Tenía un MTBF de 30 minutos.



IBM 704

Fue la primer computadora con punto flotante y a diferencia de la IBM 701 estas si salieron a venta al mercado, además era el doble de rápida que su antecesora. Tenia dispositivos para copiar tarjetas a cinta offline. Tenía un MTBF de 8 horas y se podían correr programas en FORTRAN.



Máquina Dedicada

Para poder utilizar esta computadora era necesario reservar durante ciertas horas, la persona que la reservaba era el responsable de hacer todo el proceso manual para correr el software creado por lo tanto el proceso era muy lento y propenso a errores.

Las tareas principales de esta máquina eran:

1. Configurar todos los tableros de control
2. Colocar FORTRAN en los lectores de tarjetas
3. Colocar la fuente FORTRAN en la lectora, esto podría consistir desde una a varias pasadas
4. Si no se encontraban errores entonces el compilador perforaba el programa equivalente a ensamblador
5. Colocar Ensamblador en la lectora de tarjetas
6. Colocar el código ensamblador en la lectora de tarjetas
7. El ensamblador perfora el programa equivalente a lenguaje máquina
8. Se coloca el lenguaje máquina en la lectora de tarjetas y se corre el programa
9. Se recolectaban los resultados
10. Volver al paso 1



Ventajas de la máquina dedicada:

1. Absoluto control del programador
2. Pocos expertos al inicio
3. Fácil de establecer responsabilidades y controles

Desventajas de la máquina dedicada:

1. Trabajo repetido entre un usuario y otro
2. No hay estándares
3. Los programadores debían ser expertos en hardware
4. Uso muy ineficiente de recursos

Para agilizar todo el proceso que se debía realizar por estas máquinas se implementaron las siguientes cosas:

1. Se creó el administrador único para cada computadora, el cual iba a realizar todo el trabajo, basado en una idea administrativa
2. Al haber un operador único era más fácil recolectar todos los trabajos y factorizar los pasos
3. Se crean configuraciones estándar para cada máquina
4. Se crea el procesamiento en lotes o **procesamiento Batch**
5. Se incrementó el **throughput** (Cantidad de trabajos terminados por unidades de tiempo) de los centros de cómputo

Productividad del Centro de Cómputo:



¿Se necesita un operador?

La idea de un operador funcionó demasiado bien, pero de seguro había una forma de mejorar este proceso ya que su trabajo era muy mecánico. Así que surgió la pregunta sobre cómo deshacerse de él y la respuesta más lógica fue 'Usando software'.

--- FIN ---

Lecturas de la semana

- | | |
|------|--|
| 0101 | General Motors/North American Monitor for the IBM 704 Computer |
| 0103 | Interrupts, Traps, and Exceptions |
| 0106 | Procedure calls, Interrupts, and Exceptions |
| 0107 | Interrupts |