

Instituto Tecnológico de Costa Rica  
Escuela de Computación  
Compiladores e Intérpretes  
Grupo:40  
Profesor: Francisco Torres  
Apuntes de la materia vista el 15 de febrero del  
2017

Dylan Rodríguez Barboza. 2015057714

February 18, 2017

## Contenidos

1 Quiz 1	3
2 Niveles de máquinas vistos en clase	4
3 Lenguajes de alto nivel	4
4 IBM 701	4
5 IBM 704	5
6 Máquina dedicada	5
7 Mejora de la máquina dedicada (batch con operador)	6
8 Sistemas operativos primitivos	6
9 Job Control Language (JCL)	7
10 Entrada/Salida primitiva	8
11 Entrada/Salida más sistema operativo	8
12 Servicio del sistema operativo	8

<b>13 Definiciones vistas en clase</b>	<b>9</b>
<b>14 Sistema operativo como máquina virtual</b>	<b>10</b>

## 1 Quiz 1

Defina detalladamente los siguientes conceptos:

- ENIAC.
- John Backus.
- "First Draft" de Von Neumann.
- Maurice Wilkes.
- Ensamblador.
- Grace Murray Hopper.
- Microprogramación.
- Productividad.
- John Eckert.
- Arquitectura.

## 2 Niveles de máquinas vistos en clase

- Lenguaje de alto nivel.
- Ensamblador.
- Lenguaje máquina.
- Microprogramación.
- Hardware.

Si se toma en cuenta desde el último nivel hasta el primero, se puede notar que se gana productividad, pero se baja eficiencia, en los únicos niveles donde existe la misma eficiencia es en el ensamblador y en el lenguaje máquina.

## 3 Lenguajes de alto nivel

Independiente del hardware, eran portables, se daba la abstracción de programación, en ese entonces, los programadores pensaban en FORTRAN, tenían gran proliferación, era mucho más fácil programar. Los lenguajes de alto nivel existen desde el año 1956. Antes de esto, había un compilador para cada máquina en ensamblador.

## 4 IBM 701

Es la primera computadora comercial (en abril del año 1952). En ese entonces, IBM dominaba el mercado computacional, solo se fabricaron 19. La configuración de la IBM 701 incluía: 4096 palabras de 36 bits, 150 000 instrucciones por segundo, 150 tarjetas por segundo (entrada) y 100 tarjetas por segundo (salida). En ese tiempo, se alquilaba a \$23 750 al mes (\$217 058.96 actualmente). Estas computadoras ya utilizaban la arquitectura de Von Neumann. Utilizó tubos William (usado para almacenar electrónicamente datos binarios). Tenían un MTBF de 30 minutos. Debido a que se decía que IBM dominaba el mercado, existían impresoras, perforadoras y computadoras marca IBM.



Figura 1: IBM701

## 5 IBM 704

Fue la primera computadora con punto flotante (entre los años 1954-1960). Se vendieron o alquilaron unas 200 máquinas. El alquiler era de \$35 550 al mes en esa época (\$321 635.65 actualmente). Era el doble de rápida que la IBM 701, sin embargo, era incompatible con su predecesora. Manejaba dispositivos para copiar tarjetas a cinta off-line. Tenía una memoria de núcleos de ferrita (para el filtro de corrientes parásitas). Tenía un MTBF de 8 horas y tenía FORTRAN disponible. También se tenía LISP para esta computadora, además del programa MUSIC 1 (primer programa de música para computadora).



Figura 2: IBM704

## 6 Máquina dedicada

Era el uso típico de computadoras en los 1950's. Se reservaban horas de máquina. El programador, científico o ingeniero era la persona responsable de hacer todo. Se hacía la secuencia del boot manual, por lo que era lenta y propensa a errores. El caso ideal para su uso era el siguiente:

- Configurar los tableros de control.
- Colocar FORTRAN en la lectora de tarjetas. Dar boot al computador; ya la computadora estará lista para compilar FORTRAN.
- Colocar fuente FORTRAN en la lectora. Para leer a veces se debían hacer varias pasadas.
- Si no hay errores, el compilador perfora el programa equivalente a ensamblador.
- Colocar ensamblador generado por compilador en lectora de tarjetas, se hacen las pasadas necesarias hasta que se genere.
- Colocar lenguaje máquina en lectora de tarjetas. Dar boot al computador, luego, el programa correrá.
- Recolectar el listado de impresora.

Luego de estos pasos, llegará el siguiente usuario y hace los pasos hechos antes. Las ventajas encontradas fueron: más dinero para IBM. Era más rápido que hacerlo “a mano”. Se tenía un centro computarizado disponible (no tenía que perforar en otro lugar, ya que se tenía una perforadora como parte del equipo). Entre otras ventajas el programador tenía el control absoluto, era fácil establecer responsabilidades y controles. Las desventajas encontradas fueron: trabajo repetido entre un usuario y otro, no habían estándares, los programadores debían ser expertos en hardware, se daba un uso muy ineficiente de los recursos.

Fun fact 1: los programadores se vestían como Jorge Vargas.

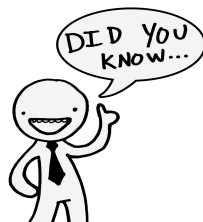


Figura 3: Fun fact 1

## 7 Mejora de la máquina dedicada (batch con operador)

Se tuvo una idea administrativa: se creó el operador de computadoras (¡Fuera programadores!). Se recolectan los trabajos y se factorizan pasos, es decir, se carga el compilador y se cargan todos los programas seguidos. Se dan las configuraciones estándar, procesamiento en lotes (batch) y un gran incremento del throughput del centro de cómputo, es decir, la cantidad de unidades terminadas por unidad de tiempo. Fue un gran aumento de productividad con respecto a la máquina dedicada. La idea del operador funcionó bien, se tenía un trabajo casi mecánico, sin embargo, se quería hacer un software que reemplazara al operador.

## 8 Sistemas operativos primitivos

En este tiempo había más compiladores disponibles, entre ellos, para FORTRAN, Cobol y Algol, menos fallas de hardware, había más periféricos. Se tenía la novedosa idea de crear un software que reemplazara al operador (el software debía seleccionar al compilador, compilar, ejecutar y guardar los resultados).

“Algo como un monitor, o sistema operador, o... sistema operativo”.

En ese tiempo eran comunes las preguntas del tipo: ¿Cómo sabe el SO hasta dónde llega cada trabajo? ¿Qué compilador usar? ¿Datos? ¿Fin de datos?



Figura 4: Compilador para COBOL

## 9 Job Control Language (JCL)

Fue la primera interfaz usuario – sistema operativo. En ese tiempo, si no se sabía JCL, no se podía utilizar el sistema operativo. Habían tarjetas intercaladas entre los trabajos de los usuarios, Si el JCL fallaba, ni siquiera se podía correr un programa. La sintaxis era primitiva. Es el antecesor a shell languages. El batch con SO representó una mayor productividad comparado con el batch con operador y la máquina dedicada.

```

EDIT MISC TEXT: PRINT-JCL                                COLUMNS 001 072
COMMAND INPUT ***                                         SKROLL *** HALF
.....1.....2.....3.....4.....5.....6.....7..
001000 //SYSCUT DD SYSCUT(*)
001100 //
001200 //
001300 //USE JWSRPT DESTINATIONS WHEN PRINTING A REPORT SERIES FROM ROLLS
001400 //
001410 //JWSRPT DD SYSCUT-A,DEST=APTR35, <---- 80 CCL LASER
001510 //JWSRPT DD SYSCUT-A,DEST=APTR35A, <---- 132 CCL LASER
001511 //JWSRPT DD SYSCUT-A,DEST=APTR35B, <---- 160 CCL LASER
001520 //JWSRPT DD DSM-CSCAFR.122D TEST, <---- TSO DATA SET -
001600 //JWSRPT DD SYSCUT-A,DEST=APTR33, <---- LINE PRINTER -
001700 //JWSRPT DD SYSCUT-A,DEST=APTR38, <---- LINE PRINTER -
001710 //JWSRPT DD SYSCUT-A,DEST=APTR38A, <---- LINE PRINTER
001720 //JWSRPT DD SYSCUT-A,DEST=APTR38B, <---- LINE PRINTER
001730 //JWSRPT DD SYSCUT-I, <---- LASER PRINTER - SIPS
001800 //JWSRPT DD SYSCUT-J, <---- LASER PRINTER - SIPS
001900 //JWSRPT DD SYSCUT-(*), <---- PRINT TO HOLD QUEUE
002000 // DCE=(ECDF44VEA,RECL=260,BKSIZE=22364),
002100 // COPIES=1 <---- CHANGE IF GT 1 COPY

ACTION: _____ 1 Help 3 End 5 Find 6 Change 7 Pg Bwd 8 Pg Fwd
|

```

Figura 5: Job Control Language (JCL)

## 10 Entrada/Salida primitiva

Estaba la Arquitectura de Von Neumann, sugiere un espacio de direcciones para E/S, había instrucciones de lenguaje máquina para ese espacio. No era fácil, llegaban dispositivos nuevos más complejos, al inicio cada uno escribía sus rutinas E/S, dichas rutinas eran pasadas entre programadores. Había bibliotecas de E/S que eran agregadas a cada programa en ensamblador.

Fun fact 2: existe un lugar místico llamado biblioteca, la cual posee elementos increíbles llamados libros, hechos de papel, que contienen información importante para la investigación.



Figura 6: Fun fact 2

## 11 Entrada/Salida más sistema operativo

Se implementó, ya que todo el mundo necesitaba hacer E/S, por lo que surgió la idea de tenerlas siempre disponibles en memoria. Los programas se comunican con el sistema operativo para pedirle operaciones de E/S.

## 12 Servicio del sistema operativo

Las operaciones de entrada y salida del sistema operativo eran facilitadas por los device drivers. El sistema operativo proporciona servicios a los programas, se daban los system calls (mecanismo utilizado por la aplicación para solicitar un servicio al sistema operativo), estas se convierten en instrucciones virtuales. El sistema operativo va creciendo en tamaño y complejidad.





Figura 7: Dispositivos de entrada y salida

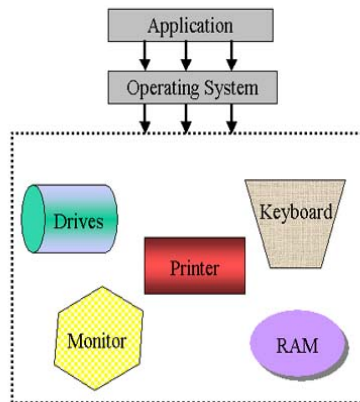


Figura 8: Sistema operativo

## 13 Definiciones vistas en clase

- Real: lo que se ve y existe.
- Virtual: lo que se ve, pero no existe.
- Transparente: lo que no se ve, pero existe.
- Máquina: dispositivo (real, virtual o transparente) que realiza una tarea computacional.

## 14 Sistema operativo como máquina virtual

Se define la máquina virtual como una máquina que se ve, pero no existe. El sistema operativo se constituye en una capa encima del hardware. El sistema operativo nos quita la máquina y a cambio nos brinda servicios. Los compiladores viven sobre la máquina virtual del sistema operativo. Es una idea tan exitosa que las capas se redefinen. El hardware se rodea del sistema operativo, el ensamblador corre sobre el sistema operativo. Las aplicaciones son hechas en alto nivel, que traduce a ensamblador que corre en un sistema operativo.



Figura 9: Máquinas virtuales sobre máquina