

## Notes from the Operative Systems

Relevants:

**Hector Robles Martinez (gh@hector290601)**

Section:  
Segunda clase



Facultad de Ingeniería  
Ingeniería en computación  
Febrary 16, 2023  
February 16, 2023

# Contents

<b>I</b>	<b>Warnings</b>	<b>2</b>
<b>1</b>	<b>Disclaimer</b>	<b>3</b>
1.1	Info source . . . . .	3
1.2	Limited Responsibility . . . . .	3
1.3	Authors List . . . . .	3
<b>II</b>	<b>Repaso histórico</b>	<b>4</b>
<b>2</b>	<b>Etapas 0</b>	<b>5</b>
2.1	Computadoras de propósito particular . . . . .	5
<b>3</b>	<b>Etapas 1</b>	<b>7</b>
3.1	Computadoras de propósito general . . . . .	7
<b>4</b>	<b>Etapas 2</b>	<b>8</b>
<b>III</b>	<b>Authors</b>	<b>9</b>
<b>5</b>	<b>Hector Robles Martínez</b>	<b>10</b>
5.1	Attributions . . . . .	10
5.2	First modify . . . . .	10
5.3	Last modify . . . . .	10
5.4	About me . . . . .	10

# Part I

## Warnings

# Chapter 1

## Disclaimer

### 1.1 Info source

Toda la información contenida en éste documento, ha sido recolectada como notas de clase, sin propósito de ser un referente ni para su consulta externa, a menos que se indique lo contrario en ésta misma sección.

### 1.2 Limited Responsibility

Como se ha mencionado antes, la información contenida son partes de las notas de clase, cualquier error favor de comunicarse con el (o los) auto (o autores) del documento.

### 1.3 Authors List

Si usted hace cambios, modificaciones, o cualquier alteración al contenido de éste documento, favor de añadirse en la lista de autores.

## Part II

# Repaso histórico

## Chapter 2

# Etapa 0

### 2.1 Computadoras de propósito particular

Computadoras de propósito particular, ejemplos como la pascalina, que era capaz de calcular integrales a partir de métodos numéricos

Babage nunca construye sus máquinas analíticas, pero dejó muy detallados sus diseños e ideas.

En la época e Babage no existía el clutch, eran spruckets.

En la era moderna, se han construido dos máquinas analíticas de Babage, por encargo de Microsoft, una está en la colección privada de Bill Gates, la otra está en el museo de la computación, en Silicon Valley.

Las tarjetas perforadas modernas tienen 80 columnas, de ahí salió la costumbre de dejar el código en 80 columnas. Aproximadamente en 1880, se crea un dispositivo mecánico para calcular rotores, creando cifrados como el cifrado de César, esta tecnología se desarrolla en Alemania, en la primera guerra mundial, para la segunda guerra mundial, se comienza a usar la comunicación radial, usando radio de banda corta.

La alemania Nazi comienza a usar las máquinas de rotores, con libros de claves diarias, con duración de entre una semana y un año.

A principios de la guerra, la inteligencia Belga capturó una máquina enigma, posteriormente llega Alan Turing a tratar de invertir el proceso de los rotores.

En la guerra del pacífico se usan las univacs.

Desde la época de Napoleón, acostumbraba apuntar todos los cañones al mismo lugar, para aumentar sus probabilidades de acertar los tiros, cuando la costumbre en la época era dejar que el artillero decidiera el objetivo y disparara a voluntad.

En la guerra del pacífico se caracterizaba el arma, para poder modelar matemáticamente el tiro para la trayectoria deseada, dando manuales para los artilleros donde indicaban cómo obtener de manera sencilla las trayectorias.

Deciden construir la *Mark II*, la univac usaba bulbos, pero en la guerra del pacífico se quedaron sin bulbos, se decide hacer la computadora con relevadores y mandar a gente de la marina a la universidad para poder acelerar los desarrollos, al año la *Mark II*, generó las tablas de cálculo casi al final de la guerra del pacífico. Luego se decide conectar un equipo telegráfico, donado por el telégrafo de Hawaii, de tipo telégrafo perforado.

Con la *Mark II* se inventan las subrutinas y el lenguaje ensamblador.

**Grace Cooper** Es quien da la idea de crear lenguajes de programación modernos, como *COBOL*.

## **Resumen**

- Recursos de Hardware justos para cumplir su aplicación.
- Surgen los compliadores (ensambladores) y subrutinas.
- Se termina de integrar la memoria funcional y la de datos.
- La arquitectura de Von Neumann contempla que la memoria de datos y la de almacenamiento sea del mismo tipo.
- No hay recursos ni necesidad de sistemas operativos por que se tiene un conocimiento detallado de la arquitectura hecha a la medida.
- La computadora de Von Neumann genera posibilidades de buffer overflow.

# Chapter 3

## Etapa 1

### 3.1 Computadoras de propósito general

Se busca que las computadoras sean más genéricas para todos los clientes, por que era muy caro diseñar máquinas específicas para cada cliente, era mejor una computadora genérica con mejores retornos de inversión.

Se tenían cientos a miles de Hz de velocidad de procesamiento.

Aproximadamente en el 1954-1955

Gran desarrollo tecnológico, grandes magnates de los negocios, se plantean vender los equipos a grandes empresas, con muchos trabajadores para solucionar el tema de los cálculos de nómina.

La inversión inicial se recupera en aproximadamente 5 años, el tiempo de vida promedio o esperado de éstas computadoras era de 10 años, dando aproximadamente 5 años de ganancia pura.

Se regalaban el lenguaje, mejoramiento de desarrollos y actualizaciones del equipo. En el segundo año se buscan nuevas cosas que vender, se comienzan a vender dispositivos teleequipos más especializados. Ya se tienen los desarrollos realizados, se comienzan a vender equipos actualizados y mejorados, es ahí cuando surgen las librerías de software para actualización y mejora continua de los que se tienen actualmente.

Comienzan los archivos de ejecución por lotes, se regalan los softwares como los Batch, que se encarga de cargar pequeños programas. El fabricante de software más importante de la historia es IBM, desde éstos años (1950s)

#### Resumen

- Computadoras con recursos fijos' "sobrados", para una aplicación.
- Se incluyen servicios de desarrollo de programas.
- Se generan nuevos lenguajes de programación de propósito específico [Cobol, Fortran, etc.]
- Se desarrollan dispositivos de I/O progresivamente mejores que requieren librerías de software para su uso.

#### Aprender PCL

#### Buscar programas de becas de PTC



## Chapter 4

# Etapa 2

Con la adopción de los transistores; con escalas de integración se tienen capacidades de hardware que crecen exponencialmente (Ley de Moore) Se comienza a tener desperdicio de capacidades de computadoras, Paul Art pide dinero a los fabricantes y desarrolladores de software, se realiza la presentación que se conoce como *La madre de las presentaciones*, propone una interfaz gráfica, el mouse, el lápiz óptico, se propone un teclado de combinaciones, con teclados estilográficos para evitar el movimiento de manos, la presentación fue grabada en cintas de video de 8mm. Se proponen novedades, como:

- Interfaz gráfica.
- Programación orientada a eventos.
- Teleconferencias.
- Primeras propuestas de internet.
- PARC - Estudios de interfaz humana.

Se genera problemas con el soporte de aplicaciones existentes.

La versión estándar de Cobol es el Cobol 86.

El mantenimiento de las aplicaciones se complica hasta llegar a la crisis del software.

*"Cada bug que corriges genera más bugs que los que resuelve"*

Buscar qué son los circuitos supertransistados heterodinos.

# **Part III**

## **Authors**

## Chapter 5

# Hector Robles Martínez

Name Hector Robles Martínez

Mobile +52 5510604869

Mail [robletes062901@gmail.com](mailto:robletes062901@gmail.com), [hector.robles@daimler.com](mailto:hector.robles@daimler.com) ,

Github [hector290601](https://github.com/hector290601)

### 5.1 Attributions

First update and attributions

### 5.2 First moddify

January 31, 2023

### 5.3 Last moddify

Febraury 1, 2023

### 5.4 About me

I'm a Computer Engenieering Student, actually I'm a passant on MBA.