

1.1 Introducción del Curso

El curso tiene como objetivo el estudio de los fundamentos de la organización de un computador. Se centra en el estudio de la unidad central de procesos (**CPU**) para a partir de ello derivar su interacción con las restantes unidades en la arquitectura: buses, dispositivos de almacenamiento y dispositivos de entrada-salida

A lo largo de la exposición, se incorpora una idea jerárquica en la organización de un computador, lo que permite introducir distintos niveles de abstracción en esta organización. Desde lenguajes de alto nivel (C++, java, ect.), hasta las etapas básicas que constituyen la ejecución del microprograma, intérprete de las instrucciones de máquina de la CPU (software de más bajo nivel) e incluso llegar al nivel de implementación básica de las unidades que constituyen la CPU, utilizando circuitos combinacionales y secuenciales (**hardware**). El objetivo del curso es dar una visión funcional de cada uno de estos estratos, distinguiendo claramente el ámbito de validez de cada uno de ellos.

1.2 Nociones Básicas

La computadora es una maquina electrónica cuya funcionalidad básica es la ejecución de instrucciones. En el nivel más bajo de la jerarquía de software (**lenguaje de máquina**), la CPU almacena y procesa información digital utilizando representaciones simbólicas binarias (trama de bits). Estos códigos binarios no sólo permiten almacenar información, si no que codificar las instrucciones y direcciones de memoria (con las cuales bloques de datos son indexados). De esta manera tanto los datos, direcciones e instrucciones son interpretados por la máquina como una trama de bits con un cierto formato.

La agrupación secuencial de instrucciones de un determinado lenguaje se llama **programa**. Estos programas no se restringen a la codificación del conjunto de instrucciones básicas ue puede ejecutar una determinada unidad de procesamiento (CPU), si no que es generalizable a las que puede ejecutar una máquina de procesamiento abstracta (**máquina virtual**).

El siguiente esquema muestra la arquitectura básica de un computador donde se ve las unidades centrales involucradas en el proceso de ejecutar instrucciones y el concepto del programa almacenado, impuesto por **Von Neumann**.

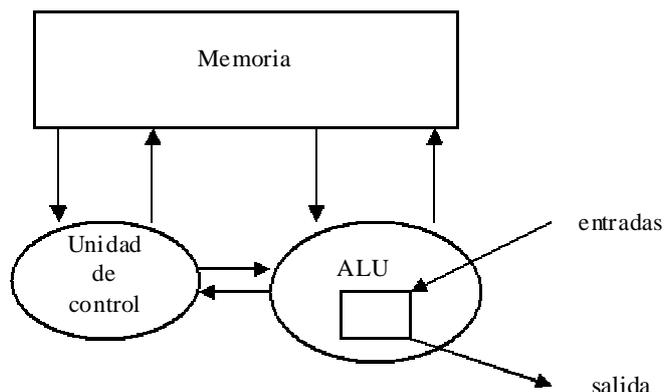


Figura 1: Esquema de arquitectura básica de Von Neumann con sus unidades elementales

Derivado de esta funcionalidad básica que es la ejecución de instrucciones, nace la necesidad de tener dispositivos de entradas y salidas (que tengan incidencia en el entorno) y dispositivos de almacenamiento de información.

1.2 Introducción a Máquina Multinivel (Jerarquía de Software)

Las instrucciones que puede ejecutar una CPU son un conjunto de secuencias binarias con un formato específico, diseñado exclusivamente para tal unidad. Dicho lenguaje de máquina está ligado íntimamente al desarrollo tecnológico de una CPU y explica por qué, un programa ejecutable de un Power-PC de 32bits no pueda ser ejecutado en un Pentium IV de 32bits. El hecho que un programador deba conocer la codificación de dichas instrucciones (para cada desarrollo tecnológico), está lejos de ser una manera amigable y flexible de desarrollar aplicaciones. Nace la necesidad de tener **estándares funcionales** de operaciones que por un lado, tengan una sintaxis más amigable para el entendimiento humano, y que a la vez, abstraigan al programador de la manera en que ellas son llevadas a cabo por una CPU en particular (**software de alto nivel**).

Generalizando el punto anterior, se pueden determinar varios niveles de abstracción (concepto de **jerarquía de software**), donde en cada nivel superior se aprecia un lenguaje más sofisticado (en términos del número de instrucciones y su complejidad) que el nivel precedente. Además cada nivel en dicha jerarquía debe estar construido sobre la base de sus precedentes. La idea es que los niveles inferiores presten servicios a un nivel superior, o en otras palabras, dicho nivel superior se construye sobre la base de las funcionalidades (instrucciones) de niveles inferiores. Esto independiza a los programadores de alto nivel de la forma en que sus programas son llevados a cabo en los niveles inferiores, y en particular a nivel de lenguaje de máquina o **programa ejecutable**.

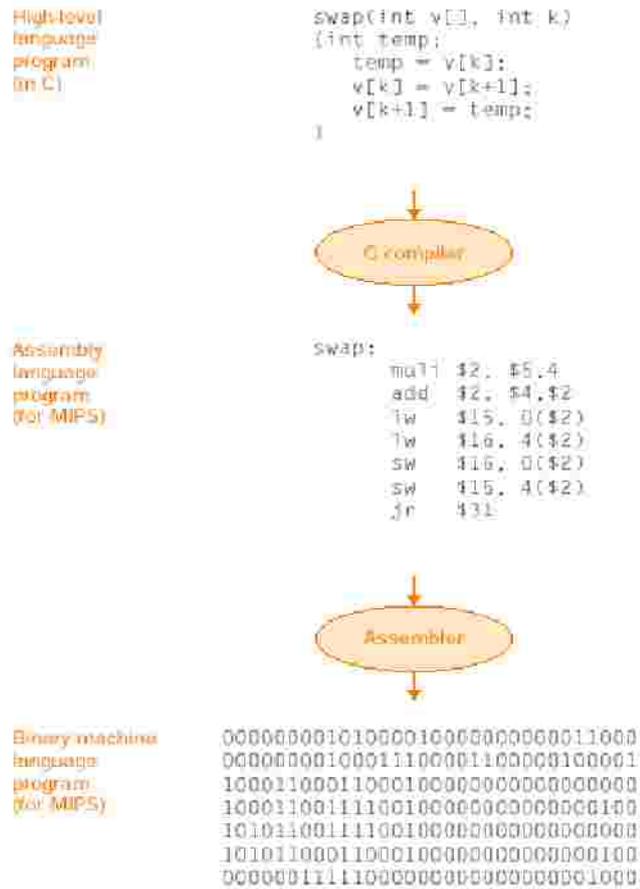


Figura 2: Diagrama de jerarquía de software

Por otro lado, este principio de abstracción se puede extrapolar a niveles más elementales que la codificación de las instrucciones básicas del procesador (lenguaje de máquina o nivel de máquina convencional). Por ejemplo se puede llevar al ámbito de como una instrucción de máquina es ejecutada por el procesador, las entidades funcionales involucrados en dicho proceso, e incluso llegar a nivel de hardware y ver aspectos de la propagación de señales y las tecnologías involucradas en las compuertas lógicas y dispositivos de almacenamiento, etc. En este punto nos encontramos en la frontera entre el **hardware** (procesador, buses, memoria, dispositivos de I/O, multiplexores, codificadores, relojes, maquinas de estado, etc.) las unidades tangibles presentes en la organización, y el software (de nivel de lenguaje de máquina).

Una visión más completa de los niveles que componen la organización de un computador es:

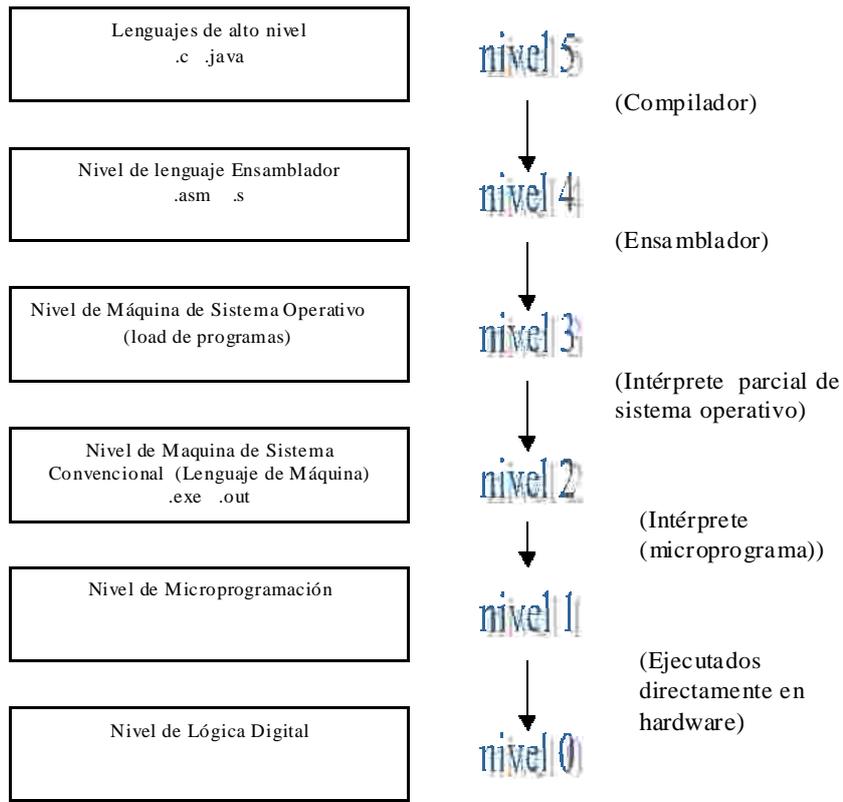


Figura 3: Detalle de los niveles de abstracción presentes en las organizaciones de computadores modernos

Nivel 0: Asociado a los aspectos de los dispositivos de hardware involucrados en la arquitectura de un computador (compuertas lógicas, circuitos integrado , tecnologías de dispositivos de almacenamiento, flip-flop, disco magnéticos, buses, dispositivos de entrada salida).

Nivel 1: En este nivel están las funcionalidades básicas de la C U (operaciones aritméticas elementales, lógicas, manipulación de bits, desplazamientos de bits) y la manera en que sus unidades elementales se sincronizan en la transferencia de información (trayectorias de datos o **microinstrucciones**) para ejecutar las instrucciones de nivel 2. En este nivel corre el intérprete del lenguaje de máquina de la arquitectura (o **microprograma**).

Nivel 2: Corresponde al conjunto de instrucciones de máquina que provee la arquitectura, (lenguaje de máquina convencional). Son las **instrucciones binarias más elementales** que se almacenan en memoria, para ser ejecutadas por el procesador. La CPU las interpreta por medio del **microprograma** de nivel 1, que transforma estas macroinstrucciones en los micro-eventos internos que ejecuta el procesador.

Nivel 3: En este lenguaje se incorporan las funcionalidades del sistema operativo, para el manejo de multiprocesos y administración de los recursos del sistema. Este nivel es interpretado por el S:O a los niveles de lenguaje de máquina.

Nivel 4: Nivel de lenguaje ensamblador (Assembler), es típicamente el conjunto de instrucciones de máquina que provee la arquitectura en un formato simbólico que facilita la tarea a programadores. El **ensamblador** es el programa encargado de recibir como entrada dichas instrucciones y codificarlas a nivel 3. Para posteriormente ser manejadas por el sistema operativo.

Nivel 5: Lenguajes de alto nivel orientados a resolución de problemas específicos, con sintaxis y conjunto de instrucciones sofisticadas, orientadas a facilitar el desarrollo de aplicaciones.

De esta estratificación se pueden distinguir tres ámbitos en la organización de un computador.

- Lo relacionado con el hardware (nivel 0, nivel 1), donde trabajan los diseñadores de dispositivos electrónicos.
- Lo asociado al sistema operativo (nivel 3), donde trabajan los programadores de sistemas (administrar los recursos de hardware abstrayendo a los estratos superiores de dicho manejo).
- Lo relacionado con las aplicaciones (nivel 4 y 5), donde trabajan los programadores de aplicaciones.

El hecho de plantear la organización de una computadora como un conjunto de niveles, permite identificar niveles de funcionalidad y entender de manera aislada cada uno de estos. Del punto de vista de diseño, este enfoque es sumamente útil pues permite mejorar aisladamente cada nivel sin involucrarse en todos los aspectos del sistema. Los diseñadores de cada nivel pueden trabajar con las funcionalidades que prestan los estratos inferiores, obviando su implementación.

1.2 Reseña de la evolución histórica de la Arquitectura de Computadores

- Generación 1: (1945-1955) introducción de los circuitos digitales mediante las tecnologías de bulbos.
ENIAC (1946) (Mauchley - Eckert) utilizaba codificación decimal, y programable en interruptores. Considerada la primera computadora electrónica de propósito general.

EDSAC basado en diseño realizado por Von Neumann, cuya arquitectura es utilizada hasta nuestros días. Impuso los conceptos de codificación binaria, instrucciones codificadas y leídas desde memoria (concepto de programa almacenado)
Su diseño tiene las unidades elementales que existen hasta hoy en las arquitecturas avanzadas (CPU, memoria, unidad de control)
- Generación de transistores (1955-1965)
Invención del transistor 1948 (Barden, Brattoun)
Creación PDP-8 (máquina de 12bit), incorpora un bus único de comunicación.
Creación de la 6600 de CDC (Control Data Corporation), introduce en la CPU unidades de trabajo paralelo, lo que logra aumentar el rendimiento en promedio en un orden de magnitud.
- Generación de circuitos integrados (1965-1980)
Introducción del concepto de familia de procesadores Intel 360, compatibles del punto de vista de su lenguaje de máquina. Incorpora 24 líneas de direccionamiento (16Mbyt)
- Generación de VLSI (1980-)
Advenimiento de los computadores personales, producto de la consolidación de tecnologías y desarrollo de economía de escala en dicho mercado.

1.3 Definiciones

En este punto aclararemos algunos conceptos que serán utilizados a lo largo del curso.

Arquitectura de Computador: Corresponde a los atributos de un sistema visibles para un programador de nivel 2, tienen incidencia en la ejecución lógica de un programa. Dentro de estos atributos se encuentra: número de instrucciones, representación binaria de los datos (enteros, punto fijo, punto flotante), técnicas para el direccionamiento de memoria, número de registros y función de cada uno de ellos, etc.

Organización de Computador: Corresponde a las unidades funcionales y sus enlaces, los cuales permiten implementar los requerimientos arquitectónicos (nivel 0 y nivel 1)

Estructura: La manera en que las componentes de una cierta unidad están interconectadas e interrelacionadas para implementar algún requisito funcional.

Funcionamiento: La operación de cada componente individual como parte de una estructura.

Bibliografía de Referencia:

Andrew Tanenbaum, “Organización de Computadores, un enfoque estructurado”, Cap 1.1-1.3.

William Stallings, “Organización y Arquitectura de Computadores”, Cap 1.1, 1.2, 2.1.