

# Sistemas Operativos

## Introducción

Departamento de Ingeniería en Sistemas y Computación  
Universidad Católica del Norte, Antofagasta.

- Clases:
  - Miércoles: 08.10 → 09.40, Lab. Universia
  - Jueves: 09.55 → 11.25, Lab. Robótica
- Correo: [miguel.solis@ucn.cl](mailto:miguel.solis@ucn.cl)

- 3 Pruebas ( $P_1$ ,  $P_2$  y  $P_3$ )
- 3 Tareas ( $T_i$ ,  $i \in \{1, 2, 3\}$ )
- 7 Actividades en clases
- Nota de presentación :

$$NP = \sum_{i=1}^3 0,2 \cdot P_i + 0,4 \cdot PT$$

Si rinde **5 o más** actividades en clases:

$$PT = (T_1 + T_2 + T_3 - \min(T_1, T_2, T_3)) / 2$$

Si rinde **2 o menos** actividades en clases:

$$PT = (T_1 + T_2 + T_3 - \max(T_1, T_2, T_3)) / 2$$

En otro caso:

$$PT = (T_1 + T_2 + T_3) / 3$$

- Pruebas:
  - Prueba 1: 08/11
  - Prueba 2: 20/12
  - Prueba 3: 17/01
- Tareas (**entrega online**):
  - Tarea 1: 08/11
  - Tarea 2: 20/12
  - Tarea 3: 17/01

- Operating System Concepts, 7th edition,  
*A. Silberschatz, P.B. Galvin, G. Gagne*

- Introducción
  - Definición y clasificación de SO
  - Definición y caracterización de procesos
  - Administración de recursos, comunicación con usuarios
  - Estructura en capas del SO

- Administración y sincronización de procesos concurrentes
  - Definición de concurrencia
  - Concepto de paralelismo
  - Región crítica
  - Sincronización de tareas
  - Prueba 1
  - ⋮
  - Interbloqueo
  - Presentación de casos



- Administración de recursos
- Casos de estudio Linux
- Análisis de rendimiento

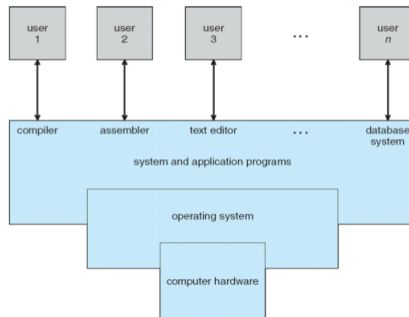
ver Programa y Planificación de la asignatura

- Programa que actúa como intermediario entre el usuario y el hardware
- **Objetivos:**
  - Ejecutar programas del usuario y solucionar problemas que puedan ocurrir
  - Hacer que un sistema computacional sea conveniente de utilizar
  - Usar el hardware computacional de una manera eficiente

- Programa que actúa como intermediario entre el usuario y el hardware
- **Objetivos:**
  - Ejecutar programas del usuario y solucionar problemas que puedan ocurrir
  - Hacer que un sistema computacional sea conveniente de utilizar
  - Usar el hardware computacional de una manera eficiente

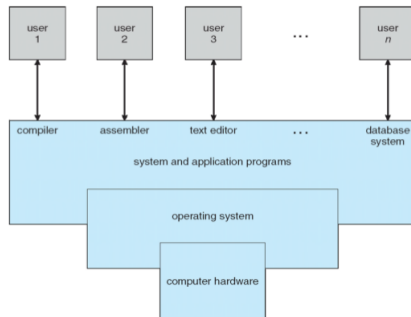
- Programa que actúa como intermediario entre el usuario y el hardware
- **Objetivos:**
  - Ejecutar programas del usuario y solucionar problemas que puedan ocurrir
  - Hacer que un sistema computacional sea conveniente de utilizar
  - Usar el hardware computacional de una manera eficiente

# Estructura de un sistema computacional



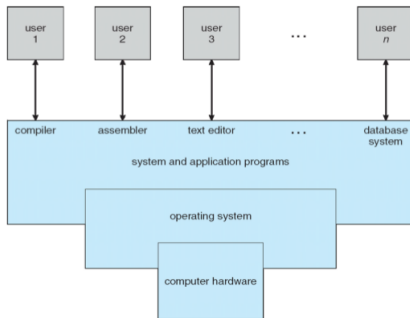
- **Hardware:** proporciona los recursos básicos (CPU, memoria, dispositivos E/S)

# Estructura de un sistema computacional



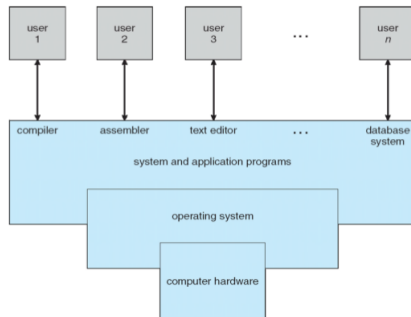
- Hardware: proporciona los recursos básicos (CPU, memoria, dispositivos E/S)

# Estructura de un sistema computacional



- **Sistema operativo:** Controla y coordina el uso de hardware entre varias aplicaciones y usuarios

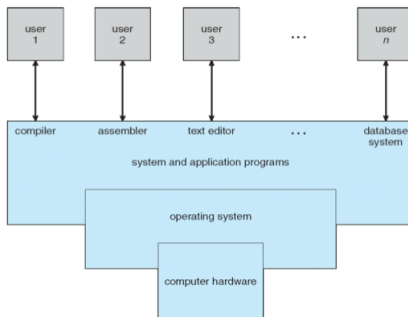
# Estructura de un sistema computacional



- Sistema operativo: Controla y coordina el uso de hardware entre varias aplicaciones y usuarios

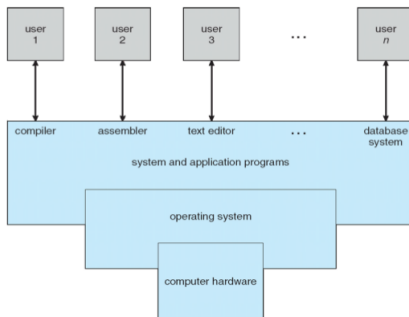


# Estructura de un sistema computacional



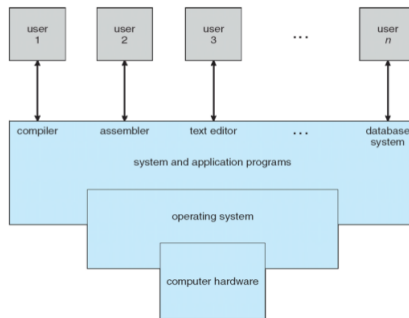
- **Aplicaciones:** Define la forma a través de la cual los recursos del sistema son utilizados para resolver los problemas de los usuarios.

# Estructura de un sistema computacional



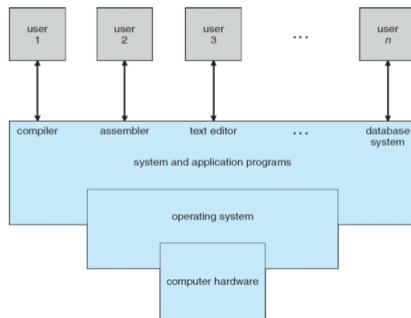
- Aplicaciones: Define la forma a través de la cual los recursos del sistema son utilizados para resolver los problemas de los usuarios.

# Estructura de un sistema computacional



- Usuarios: Personas, máquinas y otros computadores.

# Estructura de un sistema computacional



- Usuarios: Personas, máquinas y otros computadores.

# ¿Qué es un SO?

- Es un **asignador de recursos**

- Gestiona todos los recursos
- Toma decisiones entre requerimientos conflictivos para su utilización eficiente y justa

- Es un **programa de control**

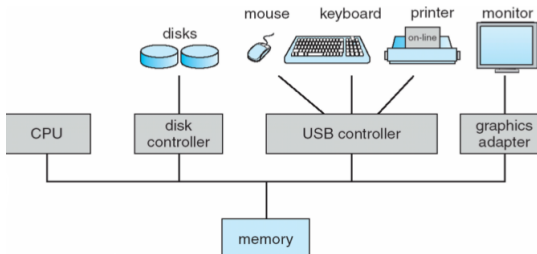
- Controla la ejecución de programas para evitar errores y usos impropios

# ¿Qué es un SO?

- Es un **asignador de recursos**
  - Gestiona todos los recursos
  - Toma decisiones entre requerimientos conflictivos para su utilización eficiente y justa
- Es un **programa de control**
  - Controla la ejecución de programas para evitar errores y usos impropios

- Es un **asignador de recursos**
  - Gestiona todos los recursos
  - Toma decisiones entre requerimientos conflictivos para su utilización eficiente y justa
- Es un **programa de control**
  - Controla la ejecución de programas para evitar errores y usos impropios

# Organización de un sistema computacional

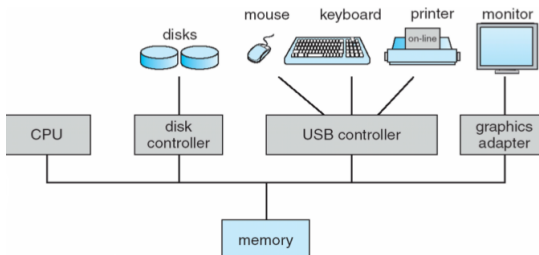


La operación requiere

- Una o más CPU y controladores con acceso a la memoria a través de un bus común



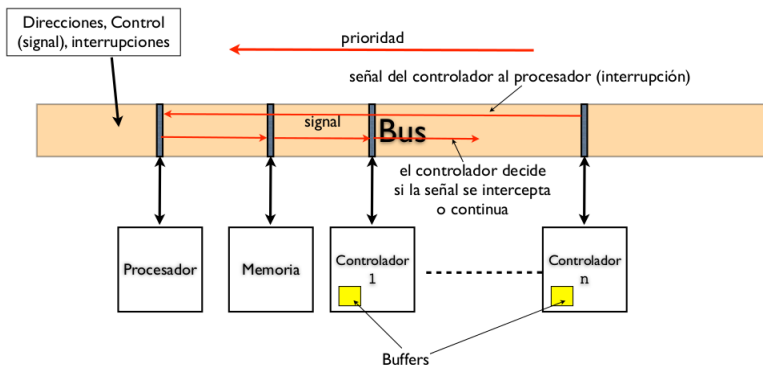
# Organización de un sistema computacional



La operación requiere

- Ejecución *concurrente* de CPU, y que controladores compitan por ciclos de memoria

# El bus que interconecta controladores



- La E/S se realiza sólo entre el dispositivo y el buffer local del controlador
- El controlador del dispositivo informa a la CPU que la transferencia a finalizado a través de una **interrupción**

- La E/S se realiza sólo entre el dispositivo y el buffer local del controlador
- El controlador del dispositivo informa a la CPU que la transferencia a finalizado a través de una **interrupción**

- Es una llamada no planificada a una subrutina
- El control se transfiere a una rutina de servicio, normalmente a través del *vector de interrupción*
- El vector de interrupción contiene la dirección de la rutina de servicio

- Es una llamada no planificada a una subrutina
- El control se transfiere a una rutina de servicio, normalmente a través del *vector de interrupción*
- El vector de interrupción contiene la dirección de la rutina de servicio

- Es una llamada no planificada a una subrutina
- El control se transfiere a una rutina de servicio, normalmente a través del *vector de interrupción*
- El vector de interrupción contiene la dirección de la rutina de servicio

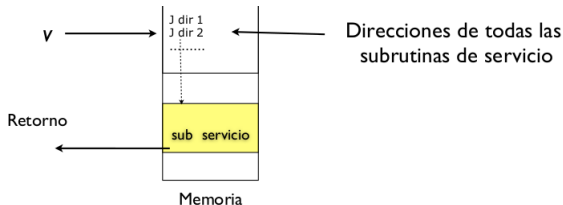
- **Trap: interrupción generada por software (por error o por requerimiento)**
- Un SO es conducido por interrupciones (cualquiera de sus acciones tiene como causa una interrupción)
- El SO preserva el estado de la CPU almacenando sus registros en un stack



- Trap: interrupción generada por software (por error o por requerimiento)
- Un SO es conducido por interrupciones (cualquiera de sus acciones tiene como causa una interrupción)
- El SO preserva el estado de la CPU almacenando sus registros en un stack

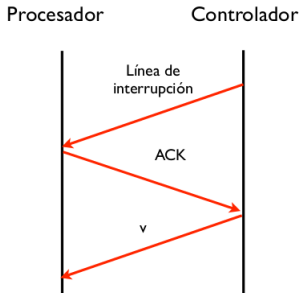
- Trap: interrupción generada por software (por error o por requerimiento)
- Un SO es conducido por interrupciones (cualquiera de sus acciones tiene como causa una interrupción)
- El SO preserva el estado de la CPU almacenando sus registros en un stack

# Interrupciones vectorizadas



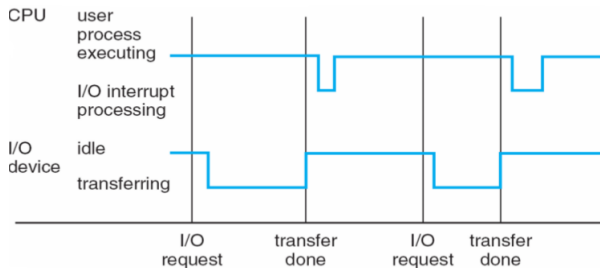
Al procesar una interrupción, se desactivan otras interrupciones para prevenir pérdidas de interrupción

# Interrupciones



v: interrupt vector address

# Interrupciones



- I/O sincrónico: Una vez que se activa el I/O, el control retorna al usuario cuando el I/O termina.
  - Sólo un I/O a la vez, no es posible concurrencia.
  - Una instrucción wait deja la CPU ociosa hasta la próxima interrupción

- I/O sincrónico: Una vez que se activa el I/O, el control retorna al usuario cuando el I/O termina.
  - Sólo un I/O a la vez, no es posible concurrencia.
  - Una instrucción wait deja la CPU ociosa hasta la próxima interrupción

- I/O sincrónico: Una vez que se activa el I/O, el control retorna al usuario cuando el I/O termina.
  - Sólo un I/O a la vez, no es posible concurrencia.
  - Una instrucción wait deja la CPU ociosa hasta la próxima interrupción



- I/O asincrónico: Una vez que se activa el I/O, el control retorna al usuario sin esperar término de I/O.
  - Se inicia mediante una llamada al sistema.
  - El SO indexa la Tabla de estado de dispositivos para saber su estado y modificarla
  - Esta tabla contiene una entrada para cada dispositivo de I/O indicando tipo, dirección y estado

- I/O asincrónico: Una vez que se activa el I/O, el control retorna al usuario sin esperar término de I/O.
  - Se inicia mediante una llamada al sistema.
  - El SO indexa la Tabla de estado de dispositivos para saber su estado y modificarla
  - Esta tabla contiene una entrada para cada dispositivo de I/O indicando tipo, dirección y estado

- I/O asincrónico: Una vez que se activa el I/O, el control retorna al usuario sin esperar término de I/O.
  - Se inicia mediante una llamada al sistema.
  - El SO indexa la Tabla de estado de dispositivos para saber su estado y modificarla
  - Esta tabla contiene una entrada para cada dispositivo de I/O indicando tipo, dirección y estado

- I/O asincrónico: Una vez que se activa el I/O, el control retorna al usuario sin esperar término de I/O.
  - Se inicia mediante una llamada al sistema.
  - El SO indexa la Tabla de estado de dispositivos para saber su estado y modificarla
  - Esta tabla contiene una entrada para cada dispositivo de I/O indicando tipo, dirección y estado

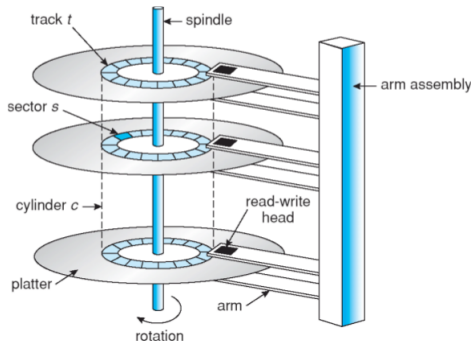
# Estructura del almacenamiento

- **Memoria principal:** arreglo de bytes, la CPU puede acceder directamente
- **Almacenamiento secundario:** extensión de la memoria principal, proporciona gran capacidad de memoria no volátil

- **Memoria principal:** arreglo de bytes, la CPU puede acceder directamente
- **Almacenamiento secundario:** extensión de la memoria principal, proporciona gran capacidad de memoria no volátil

- **Memoria principal:** arreglo de bytes, la CPU puede acceder directamente
- **Almacenamiento secundario:** extensión de la memoria principal, proporciona gran capacidad de memoria no volátil

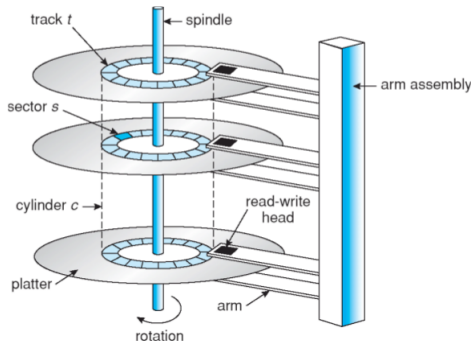
# Estructura del almacenamiento



- **Discos magnéticos:** discos rígidos cubierto de material magnético.
  - La superficie se divide en tracks, y estos en sectores

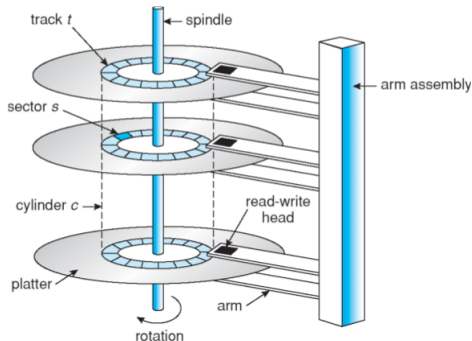


# Estructura del almacenamiento



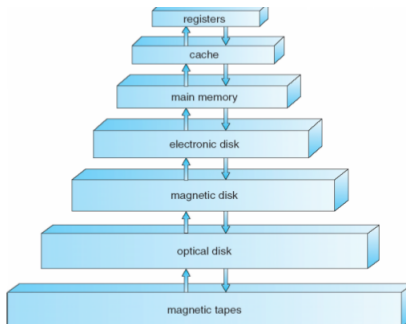
- Discos magnéticos: discos rígidos cubierto de material magnético.
  - La superficie se divide en tracks, y estos en sectores

# Estructura del almacenamiento



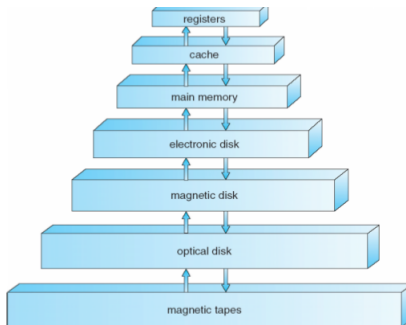
- Discos magnéticos: discos rígidos cubierto de material magnético.
  - La superficie se divide en tracks, y estos en sectores

# Jerarquía de almacenamiento



- Está determinada por velocidad de transferencia, costo y volatilidad

# Jerarquía de almacenamiento



- Está determinada por velocidad de transferencia, costo y volatilidad

- Consiste en copiar información de dispositivos lentos a dispositivos rápidos
- Memoria principal puede ser vista como una caché para almacenamiento secundario
- La información más utilizada se copia desde memorias lentas a rápidas temporalmente

- Consiste en copiar información de dispositivos lentos a dispositivos rápidos
- Memoria principal puede ser vista como una caché para almacenamiento secundario
- La información más utilizada se copia desde memorias lentas a rápidas temporalmente

- Consiste en copiar información de dispositivos lentos a dispositivos rápidos
- Memoria principal puede ser vista como una caché para almacenamiento secundario
- La información más utilizada se copia desde memorias lentas a rápidas temporalmente

- Es la memoria de de acceso rápido
- Se verifica para determinar si contiene la información correcta. Si la tiene, se usa directamente. Si no, se copian datos y después se usa.



- Es la memoria de de acceso rápido
- Se verifica para determinar si contiene la información correcta. Si la tiene, se usa directamente. Si no, se copian datos y después se usa.