Curso Sistemas Operativos

UNIVERSIDAD DE ANTIOQUIA

Clase 1 - Introducción a los Sistemas Operativos

Agenda

UNIVERSIDAD DE ANTIOQUIA

- ¿Qué hace un Sistema Operativo?
- Definición de los Sistemas Operativos
- Virtualización de Recursos
- Operaciones del sistema operativo
- Objetivos de diseño de los Sistemas Operativos

Agenda

UNIVERSIDAD DE ANTIOQUIA

- **¿Qué hace un Sistema Operativo?**
- Definición de Sistema Operativo
- Virtualización de Recursos
- Operaciones del sistema operativo
- Objetivos de diseño de los Sistemas Operativos

¿Que hace un sistema operativo?

UNIVERSIDAD DE ANTIOQUIA

¿Como se ejecuta un programa?

Para esto la CPU lleva cabo los siguientes pasos:

1. **Fetch**: Obtiene la instrucción de memoria.
2. **Decode**: Averigua la instrucción a ejecutar.
3. **Exec**: Realiza la operación indicada.
4. Se salta a la próxima instrucción y se repite el procedimiento

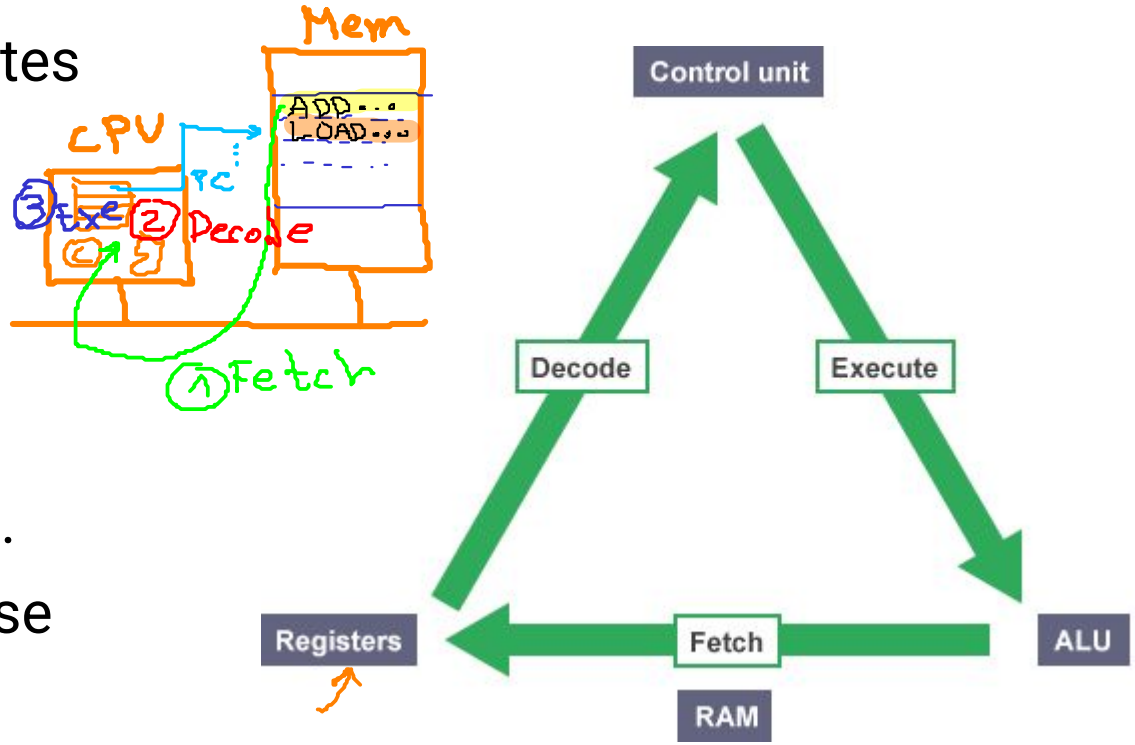


Imagen tomada de: https://www.bbc.co.uk/bitesize/guides/z2342hv/revision/5

Videos de utilidad:

- The Fetch-Execute Cycle: What's Your Computer Actually Doing? (link)
- Crafting a CPU to run Programs (link)

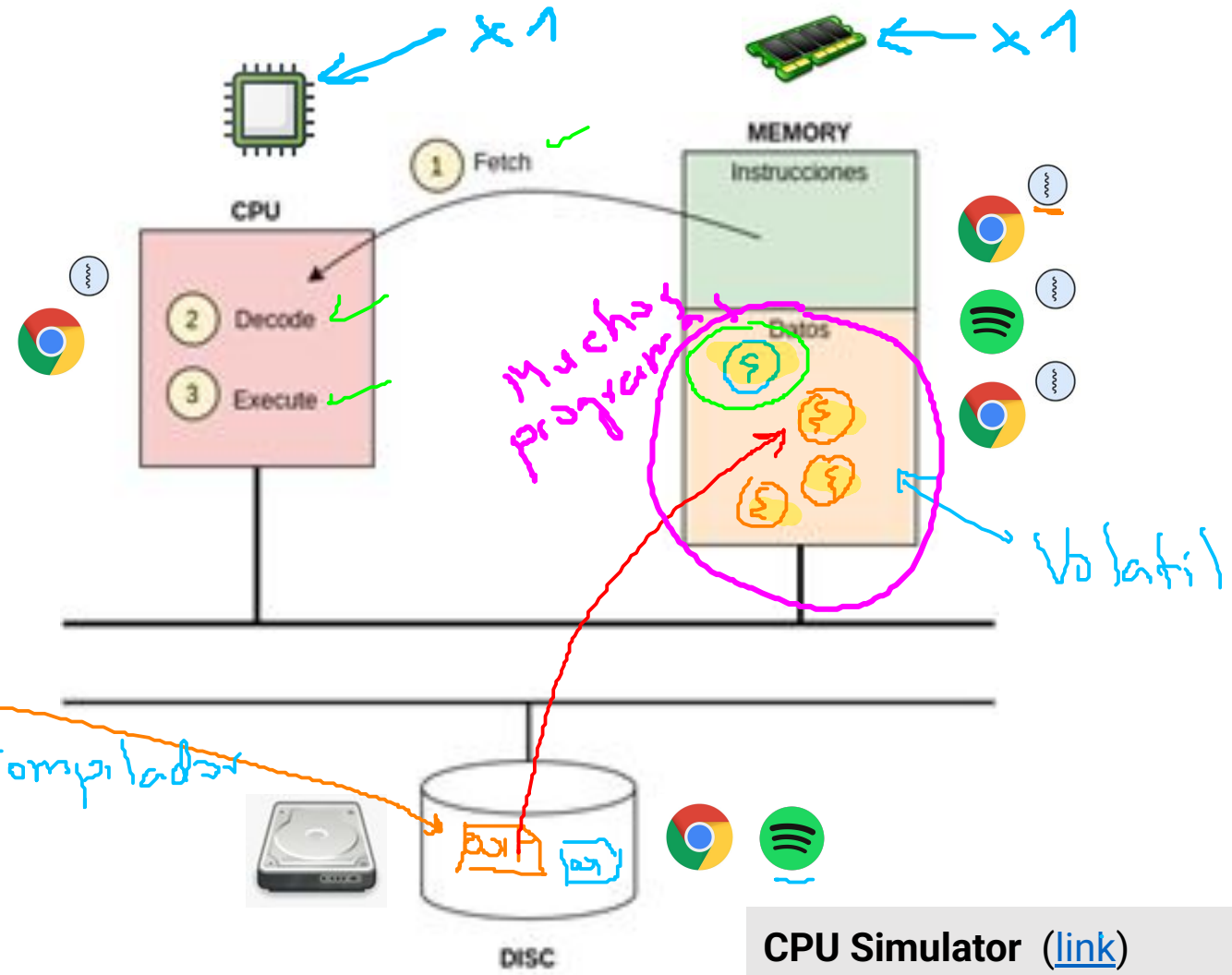
¿Que hace un sistema operativo?

UNIVERSIDAD DE ANTIOQUIA

¿Como se ejecuta un programa?

MIPS

```
.data
        x:  .word 1
        y:  .word 0
.text
main:
        lw    $t0 x
        li    $t1 2
        add   $t1 $t0 $t1
        sw $t1, y
        jr $ra
```



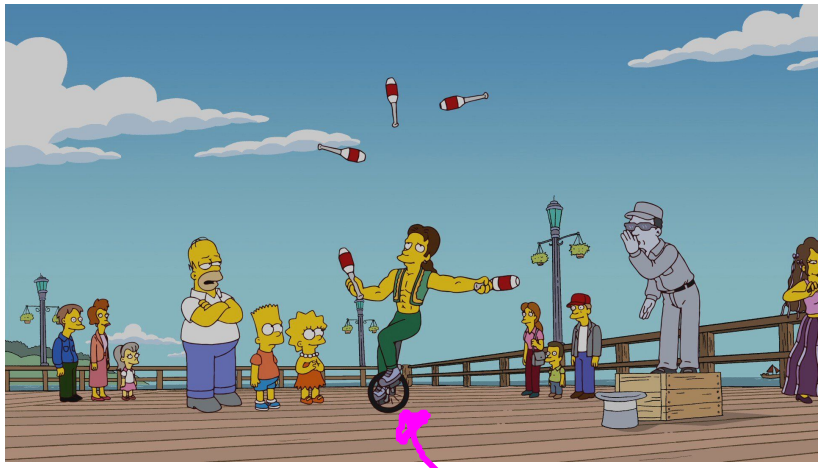
https://wepsim.github.io/

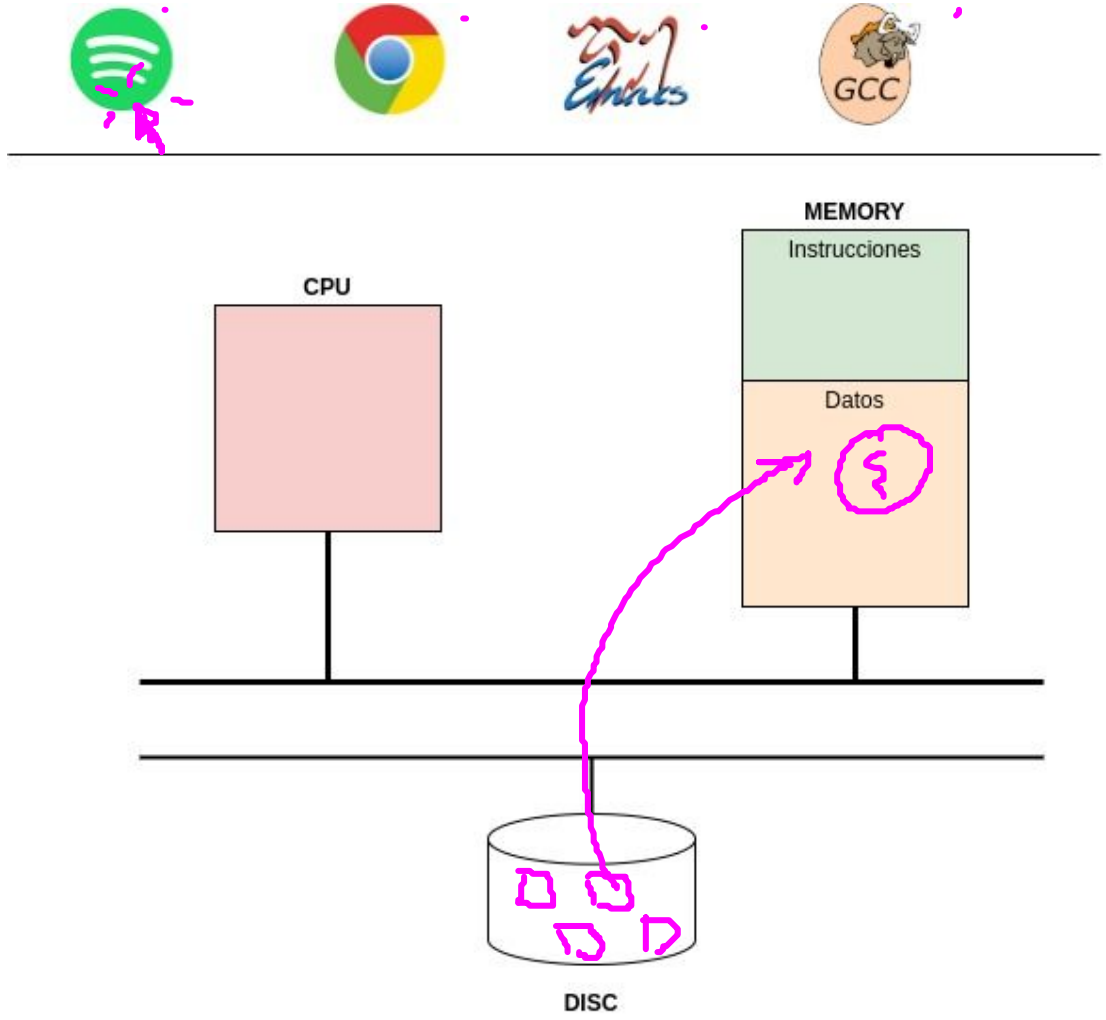
CPU Simulator (link)

¿Que hace un sistema operativo?

UNIVERSIDAD DE ANTIOQUIA

¿Como facilitar la ejecución de programas en un computador?





Incluso, ¿cómo permitir que se puedan ejecutar **varios programas** al mismo tiempo?

¿Que hace un sistema operativo?

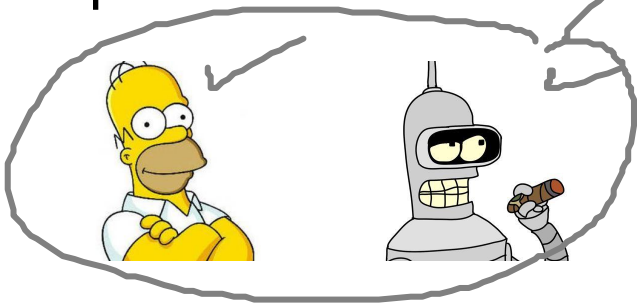
UNIVERSIDAD DE ANTIOQUIA

Estructura de los sistemas de cómputo

Componentes principales:

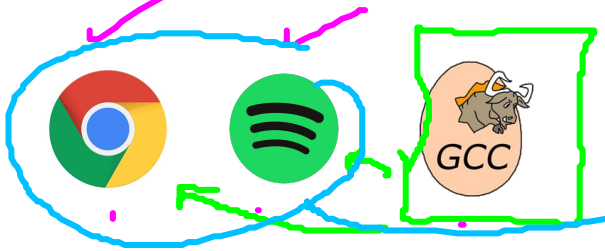
1. **Usuarios**

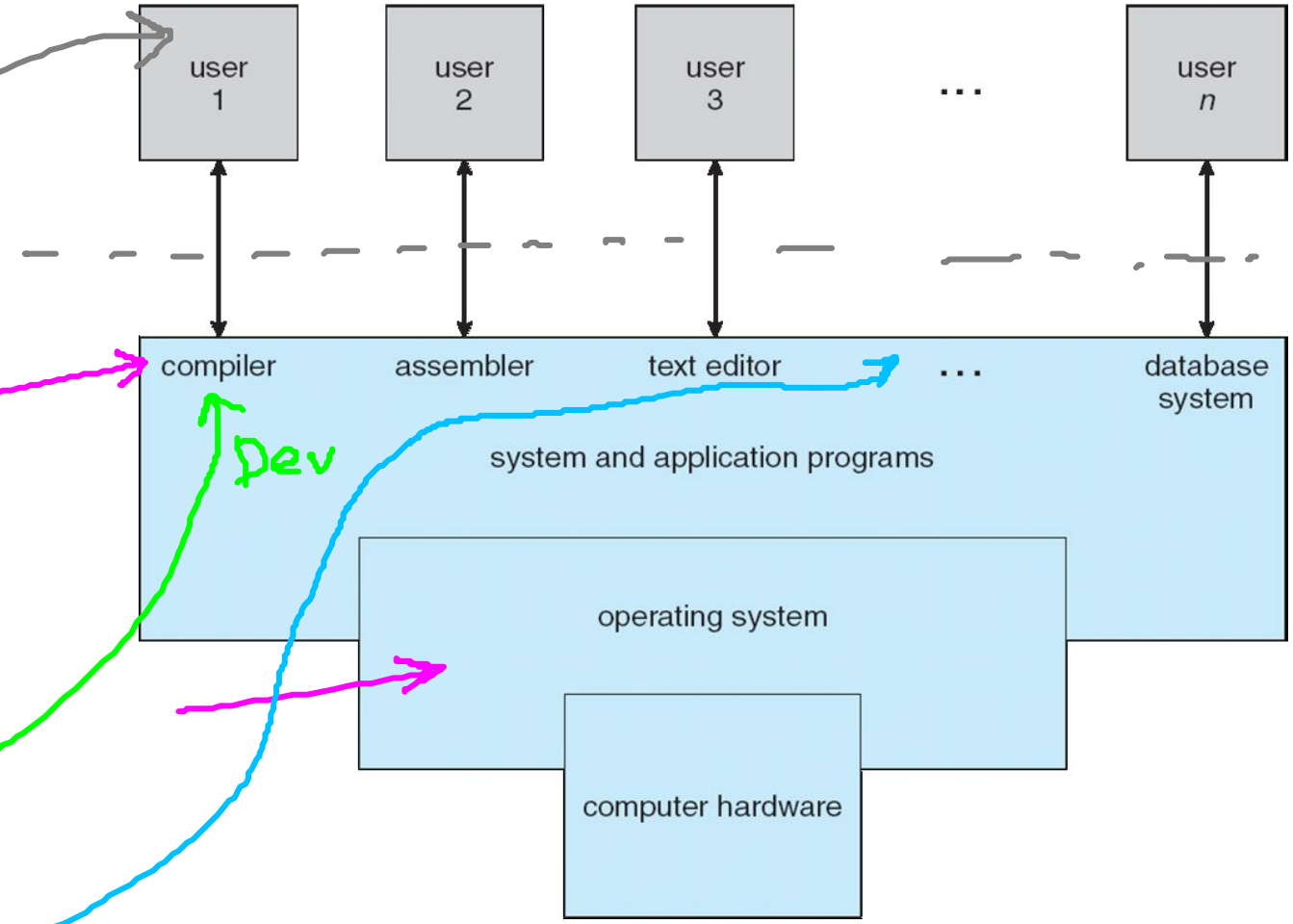
   Personas, máquinas, otras computadoras



2. **Aplicaciones**

   Compiladores, procesadores de texto, navegadores,etc.





¿Que hace un sistema operativo?

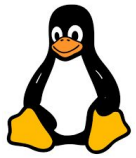
UNIVERSIDAD DE ANTIOQUIA

Estructura de los sistemas de cómputo

Componentes principales:

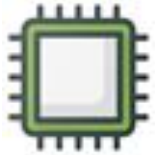
3. **Sistema Operativo**

   Controla y coordina el uso del hardware entre varias aplicaciones y usuarios

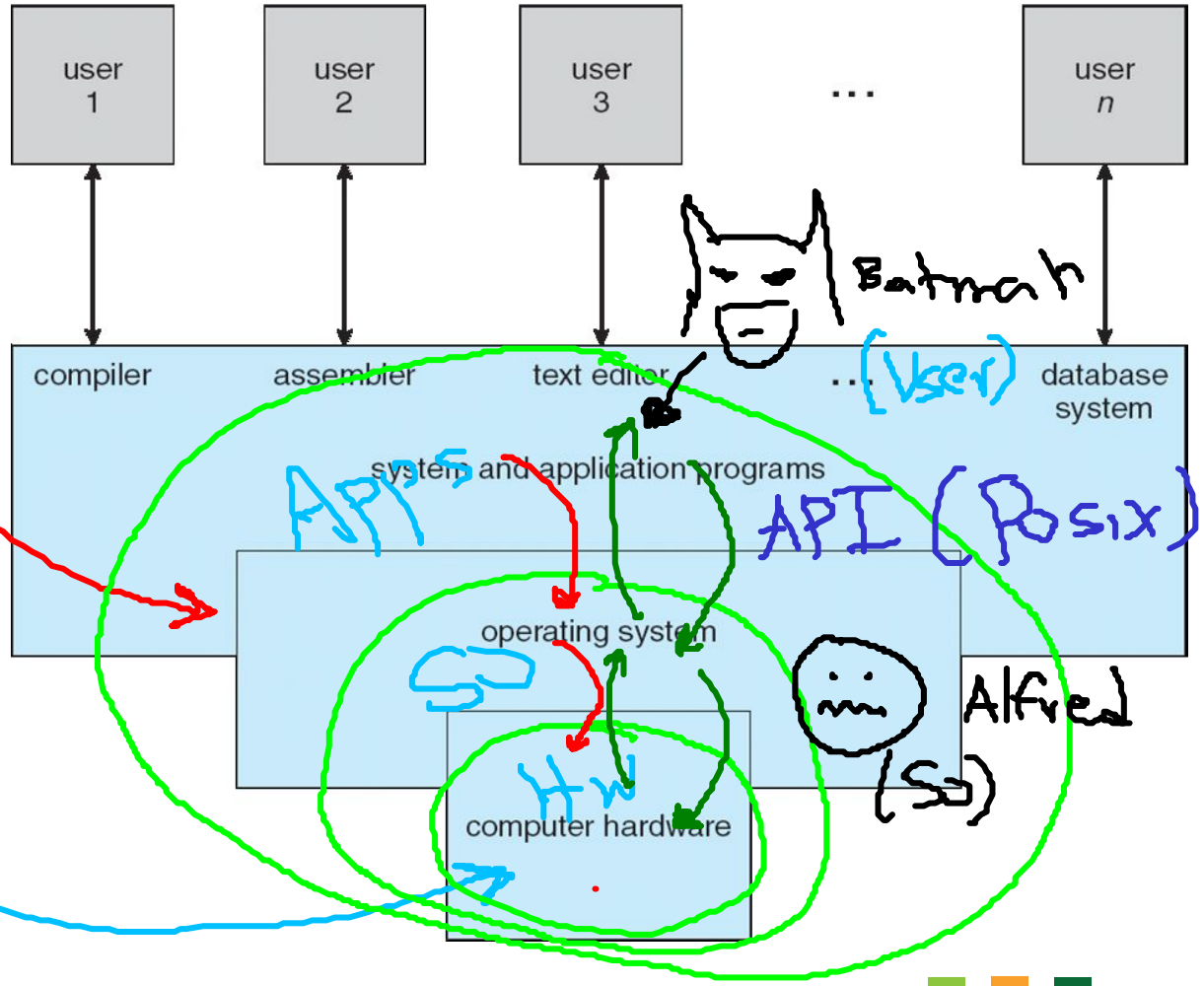


4. **Hardware**

   CPU - Memoria - Dispositivos E/S







¿Que hace un sistema operativo?

UNIVERSIDAD DE ANTIOQUIA

Clasificación de los sistemas de cómputo

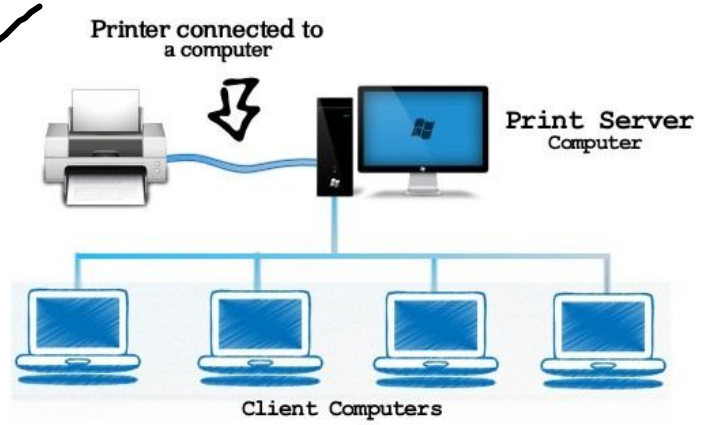
- Depende del punto de vista

Usuarios

No importa mucho la utilización de los recursos

Computadora compartida

Mantener a todos los usuarios felices



Sistema dedicado

Compromiso entre usabilidad y utilización de recursos

¿Que hace un sistema operativo?

UNIVERSIDAD DE ANTIOQUIA

Clasificación de los sistemas de cómputo

- Depende del punto de vista

Dispositivo portátil

Gestionar recursos limitados, optimizar la usabilidad

Sistemas embebidos

Interfaz de usuario muy limitada o inexistente

Agenda

UNIVERSIDAD DE ANTIOQUIA

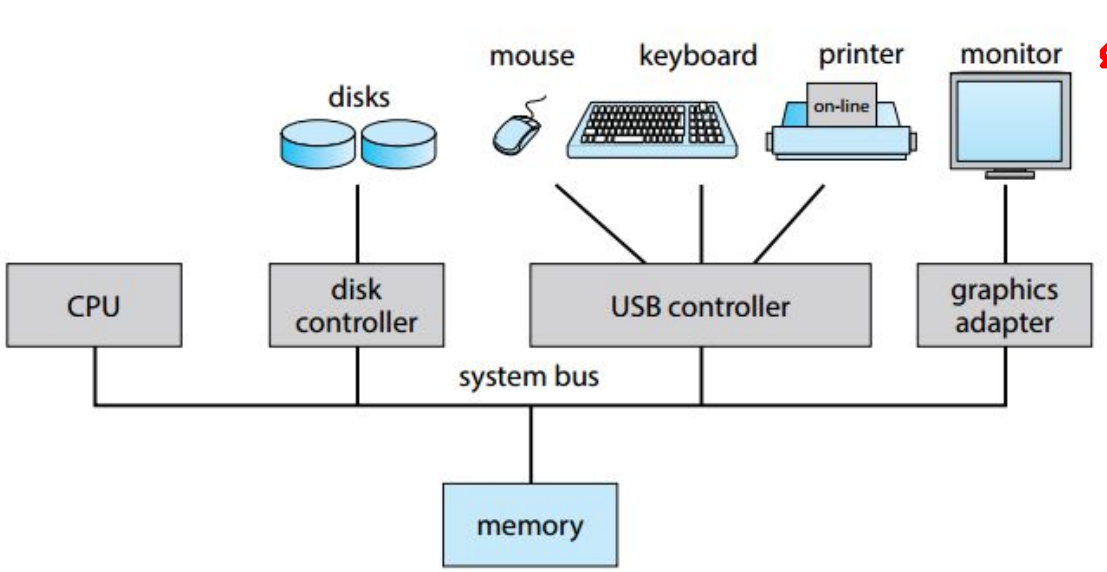
- Encuesta
- ¿Qué hace un Sistema Operativo?
- **Definición de Sistema Operativo**
- Virtualización de Recursos
- Operaciones del sistema operativo
- Objetivos de diseño de los Sistemas Operativos

Definición de Sistema Operativo

UNIVERSIDAD DE ANTIOQUIA

Es un **software** que:

1. Administrar eficiente y justa (fair) **recursos**.



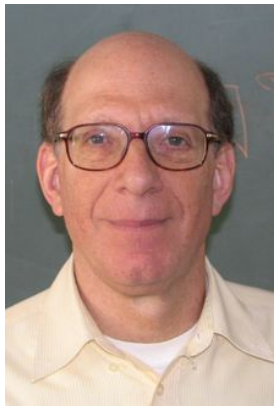
2. Se encarga de controlar la ejecución de programas
   - **Previene** errores.
   - Evita el **uso inadecuado** del sistema de computo

Imagen tomada de: Operating Systems Concepts (Silberschatz, Gagne, Galvin)

Definición de Sistema Operativo

UNIVERSIDAD DE ANTIOQUIA

No hay una definición universalmente aceptada:



***Tanenbaum*:**

An OS is a software that "performs two essentially unrelated functions: providing application programmers a clean abstract set of resources instead of the messy hardware ones and managing these hardware resources"

***Arpaci-Dusseau*:**

An OS is a body of software that "is responsible for making it easy to run programs, allowing programs to share memory, enabling programs to interact with devices, ..., as it is in charge of making sure the system operates correctly and efficiently in an easy-to-use manner"

Imagen tomada de: Operating Systems Concepts (Silberschatz, Gagne, Galvin)

Agenda

UNIVERSIDAD DE ANTIOQUIA

- Encuesta
- ¿Qué hace un Sistema Operativo?
- Definición de Sistema Operativo
- **Virtualización de Recursos**
- Operaciones del sistema operativo
- Objetivos de diseño de los Sistemas Operativos

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Sistema operativo como máquina virtual

- El sistema operativo toma un **recurso físico** y lo transforma en una **forma virtual** de sí mismo.

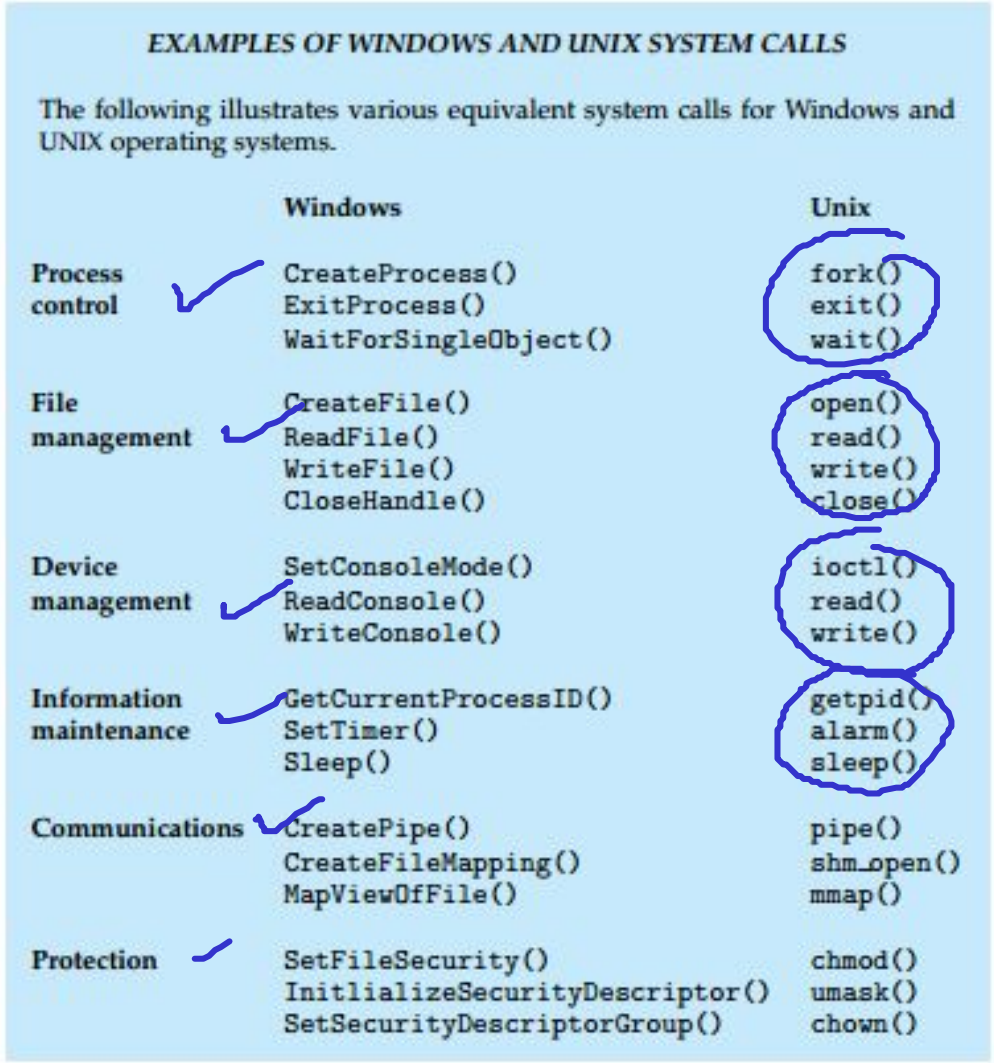
- La forma virtual es más **general**, **potente** y **fácil de usar**.

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Llamadas al sistema

- Las **llamadas al sistema** permiten que el usuario pueda **decirle** al Sistema Operativo **qué hacer**.
- El sistema operativo proporciona una interfaz (**API**, librería estándar).
- Un sistema operativo típico proporciona cientos de llamadas al sistema:
  - Ejecutar programas
  - Acceder a memoria
  - Acceder a los dispositivos
- Ejemplo: POSIX API



EXAMPLES OF WINDOWS AND UNIX SYSTEM CALLS

The following illustrates various equivalent system calls for Windows and UNIX operating systems.

	Windows	Unix
Process control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File management	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device management	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communications	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shm_open() mmap()
Protection	SetFileSecurity() InitlializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()

Imagen tomada de: Operating Systems Concepts (Silberschatz, Gagne, Galvin)

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Administrador de recursos

- El SO administra los **recursos de un sistema** de cómputo: procesador, memoria, disco duro.
- El SO permite:
  - La ejecución de múltiples programas → **compartiendo la CPU**
  - Que múltiples programas accedan simultáneamente a sus datos e instrucciones → **compartiendo la memoria**
  - Que múltiples programas accedan a los dispositivos → **compartiendo los discos**

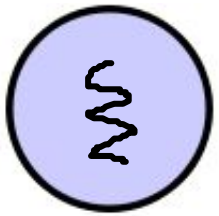
Imagen tomada de: Operating Systems Concepts (Silberschatz, Gagne, Galvin)

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Virtualización de la CPU

- El sistema tiene un gran número de CPUs virtuales.
  - Convierte una CPU en "**infinitas**" CPUs virtuales
  - Permite que múltiples programas se ejecuten "**de manera concurrente**"
- Convención



Proceso

Hilo

Archivo ejecutable

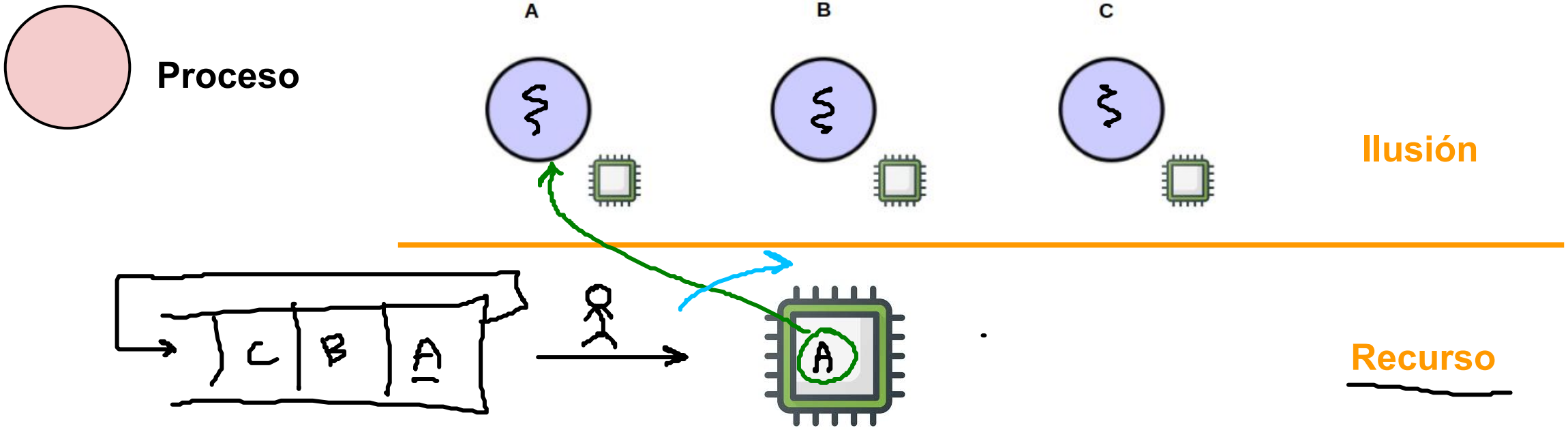
Imagen tomada de: Operating Systems Concepts (Silberschatz, Gagne, Galvin)

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Virtualización de la CPU

- El sistema tiene un gran número de CPUs virtuales.
  - Convierte una CPU en "**infinitas**" CPUs virtuales
  - Permite que múltiples programas se ejecuten "**de manera concurrente**"



Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Ejemplo

cpu.c

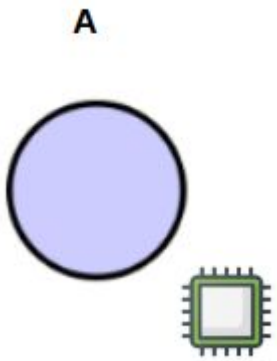
```c
#include <stdio.h>
#include <stdlib.h>
#include "common.h"


int main(int argc, char *argv[])
{
    if (argc != 2) {
     fprintf(stderr, "usage: cpu <string>\n");
     exit(1);
    }
    char *str = argv[1];


    while (1) {
     printf("%s\n", str);
     Spin(1);
    }
    return 0;
}
```

Virtualización de recursos

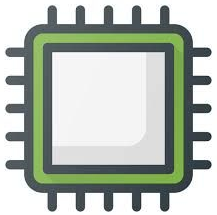
UNIVERSIDAD DE ANTIOQUIA

Virtualización de la CPU

Ilusión



Recurso



```
Dannys-MacBook-Pro:examples dannymunera$ ./a.out
usage: cpu <string>
Dannys-MacBook-Pro:examples dannymunera$ emacs 1-cpu.c
Dannys-MacBook-Pro:examples dannymunera$ ./a.out hola
hola
hola
hola
hola
hola
hola
hola
```

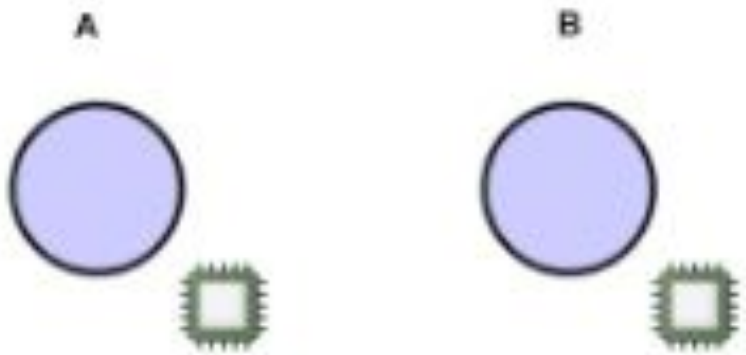
Ver video: C01P10 - Virtualización de la CPU: Ejemplos - ISI485 Sistemas Operativos (link)

Virtualización de recursos

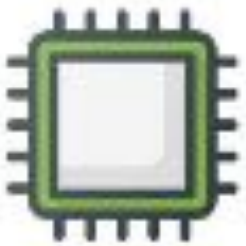
UNIVERSIDAD DE ANTIOQUIA

Virtualización de la CPU

Ilusión



Recurso



```
examples — -bash — 80×26
Dannys-MacBook-Pro:examples dannymunera$ ./a.out hola & ./a.out mundo
[1] 1301
mundo
hola
mundo
hola
mundo
hola
mundo
hola
mundo
hola
mundo
hola
mundo
hola
^C
```

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Virtualización de Memoria

- La memoria física **es un arreglo de bytes**.
- Un programa almacena todas sus estructuras de datos en la memoria:
  - Lee la memoria (**load**): Especifica una dirección para poder acceder a un dato
  - Escribe en la memoria (**write**):Especifica el dato para ser escrito en una dirección dada.

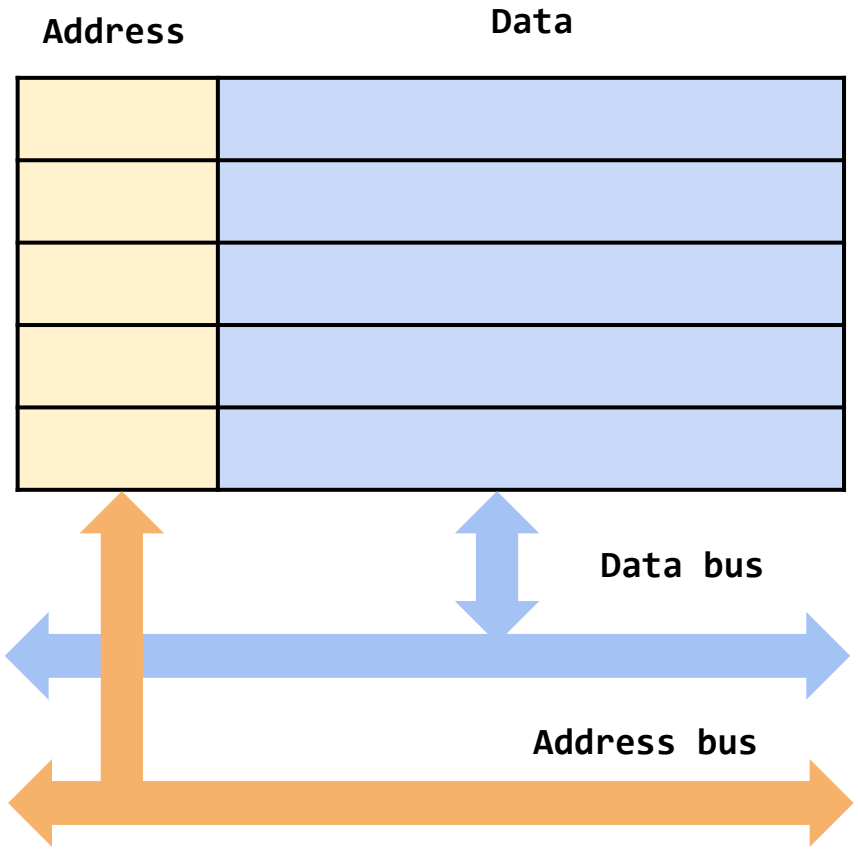


Imagen tomada de: Operating Systems Concepts (Silberschatz, Gagne, Galvin)

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Ejemplo

mem.c

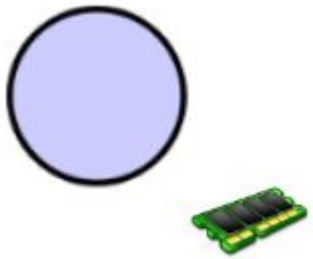
```c
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include "common.h"


int main(int argc, char *argv[]) {
    if (argc != 2) {
     fprintf(stderr, "usage: mem <value>\n");
     exit(1);
    }
    int *p;
    p = malloc(sizeof(int));
    assert(p != NULL);
    printf("(%d) addr pointed to by p: %p\n", (int) getpid(), p);
    *p = atoi(argv[1]); // assign value to addr stored in p
    while (1) {
     Spin(1);
     *p = *p + 1;
     printf("(%d) value of p: %d\n", getpid(), *p);
    }
    return 0;
}
```

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Virtualización de Memoria

Ilusión



```
Dannys-MacBook-Pro:examples dannymunera$ gcc -o 2-mem 2-
Dannys-MacBook-Pro:examples dannymunera$ ./2-mem
(1335) address pointed to by p: 0x100400190
(1335) p: 0x100400190
(1335) p: 0x100400190
(1335) p: 0x100400190
(1335) p: 0x100400190
(1335) p: 0x100400190
(1335) p: 0x100400190
```

Recurso

Ver video: C01P11 - Ejemplo Virtualización de Memoria - ISI485 Sistemas Operativos (link)

Imagen tomada de: Operating Systems Concepts (Silberschatz, Gagne, Galvin)

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Virtualización de Memoria

Ilusión

```
Dannys-MacBook-Pro:examples dannymunera$ ./2-mem & ./2-mem
[1] 1336
(1336) address pointed to by p: 0x100100080
(1337) address pointed to by p: 0x100100080
(1336) p: 0x100100080
(1337) p: 0x100100080
(1337) p: 0x100100080
(1336) p: 0x100100080
(1336) p: 0x100100080
(1337) p: 0x100100080
(1337) p: 0x100100080
```

Recurso

Imagen tomada de: Operating Systems Concepts (Silberschatz, Gagne, Galvin)

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Recurrencia

- El SO está realizando muchas cosas a la vez, primero ejecutando un proceso, luego otro, y así sucesivamente.
- Programas modernos multi-hilo presentan **problemas de concurrencia**.

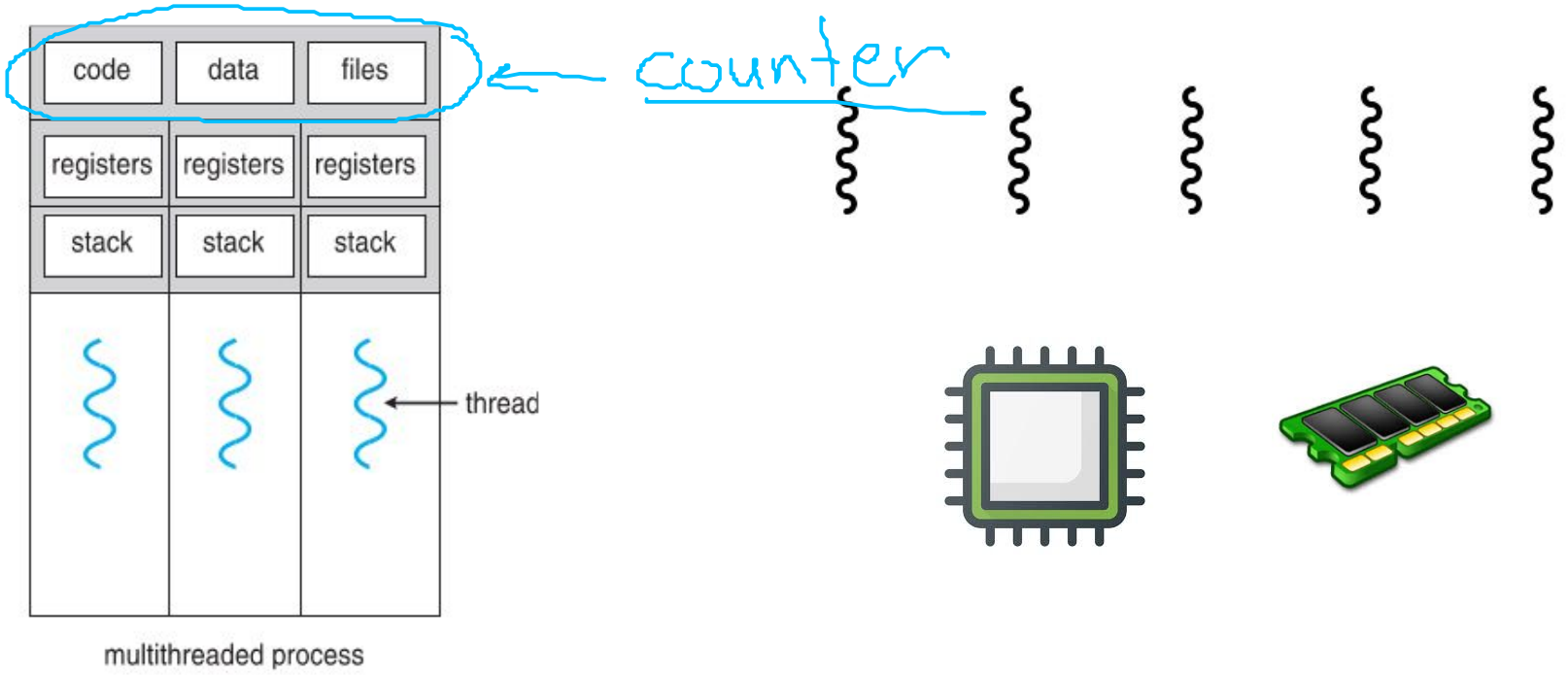


Imagen tomada de: Operating Systems Concepts (Silberschatz, Gagne, Galvin)

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Recurrencia

- **Contextualización:** Dos hilos (**threads**) comparten el balance de una cuenta en memoria). Cada hilo corre su propio código llamado **`deposit()`** tal y como se muestra a continuación.

```
void deposit (int amount) {
    balance = balance + amount;
}
```

```
load R1, balance
add R1, amount
store R1, balance
```

Responda:

- ¿Cuales son variables compartidas?
- ¿Cuales con variables privadas?

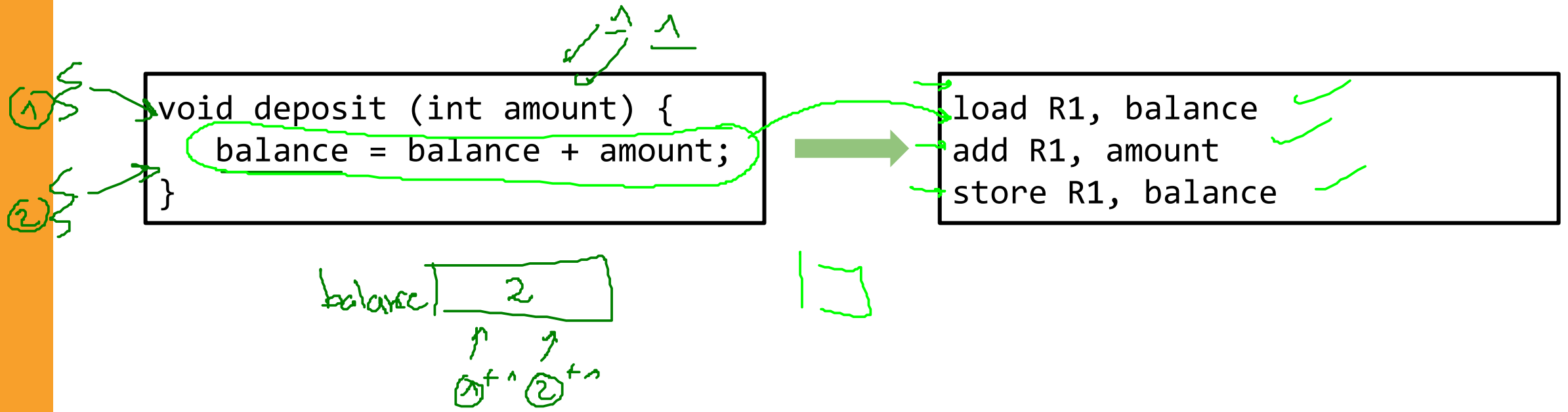
Ejemplo tomado de: http://pages.cs.wisc.edu/~remzi/Classes/537/Fall2005/Lectures/lecture4.pdf

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Recurrencia

- Incrementar una **variable compartida** usa tres instrucciones:
  - **Cargar** el valor del contador de la memoria a un registro
  - **Incrementar**
  - **Almacenar** el valor modificado en la memoria

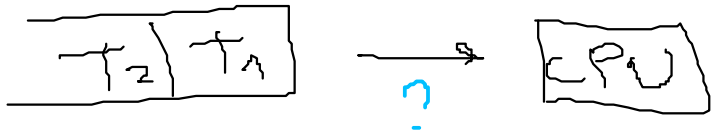


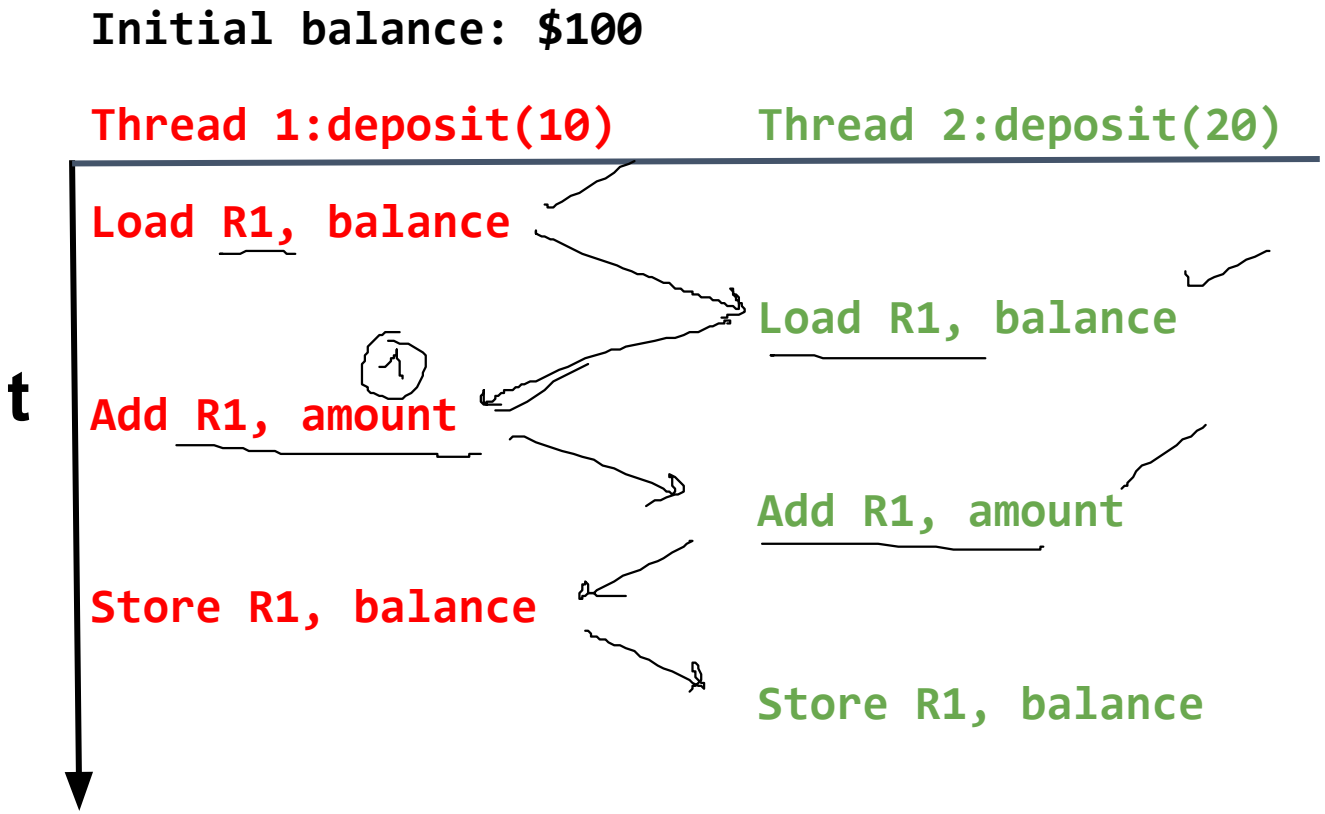
Ejemplo tomado de: http://pages.cs.wisc.edu/~remzi/Classes/537/Fall2005/Lectures/lecture4.pdf

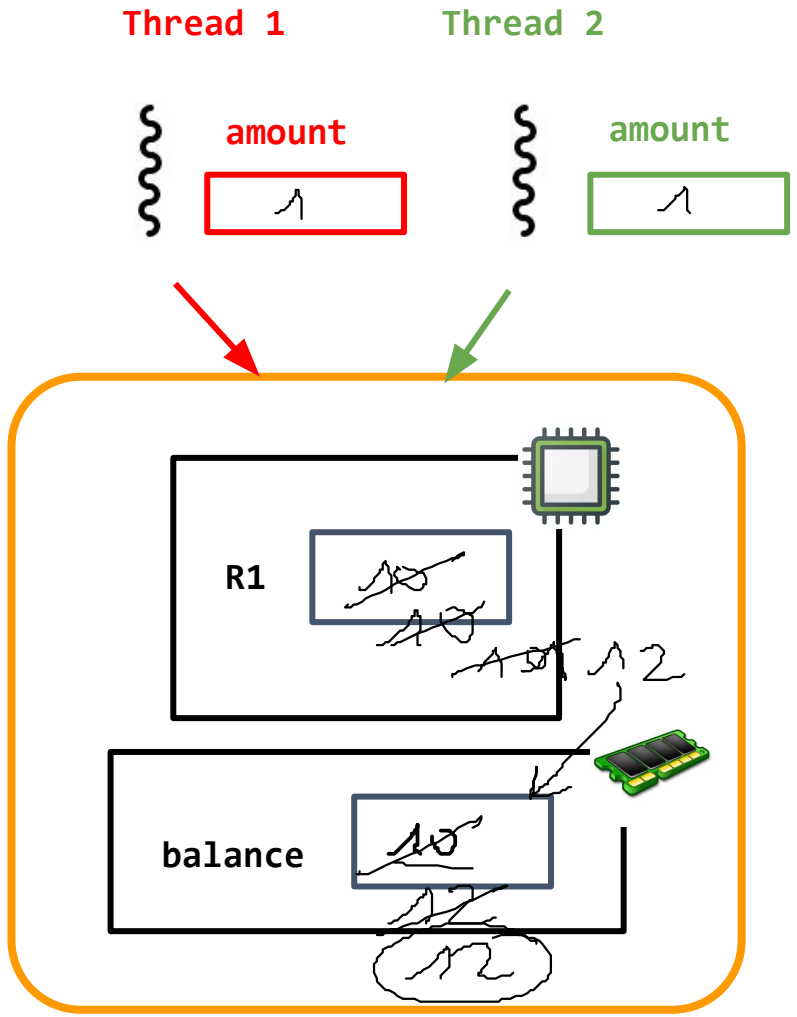
Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Recurrencia





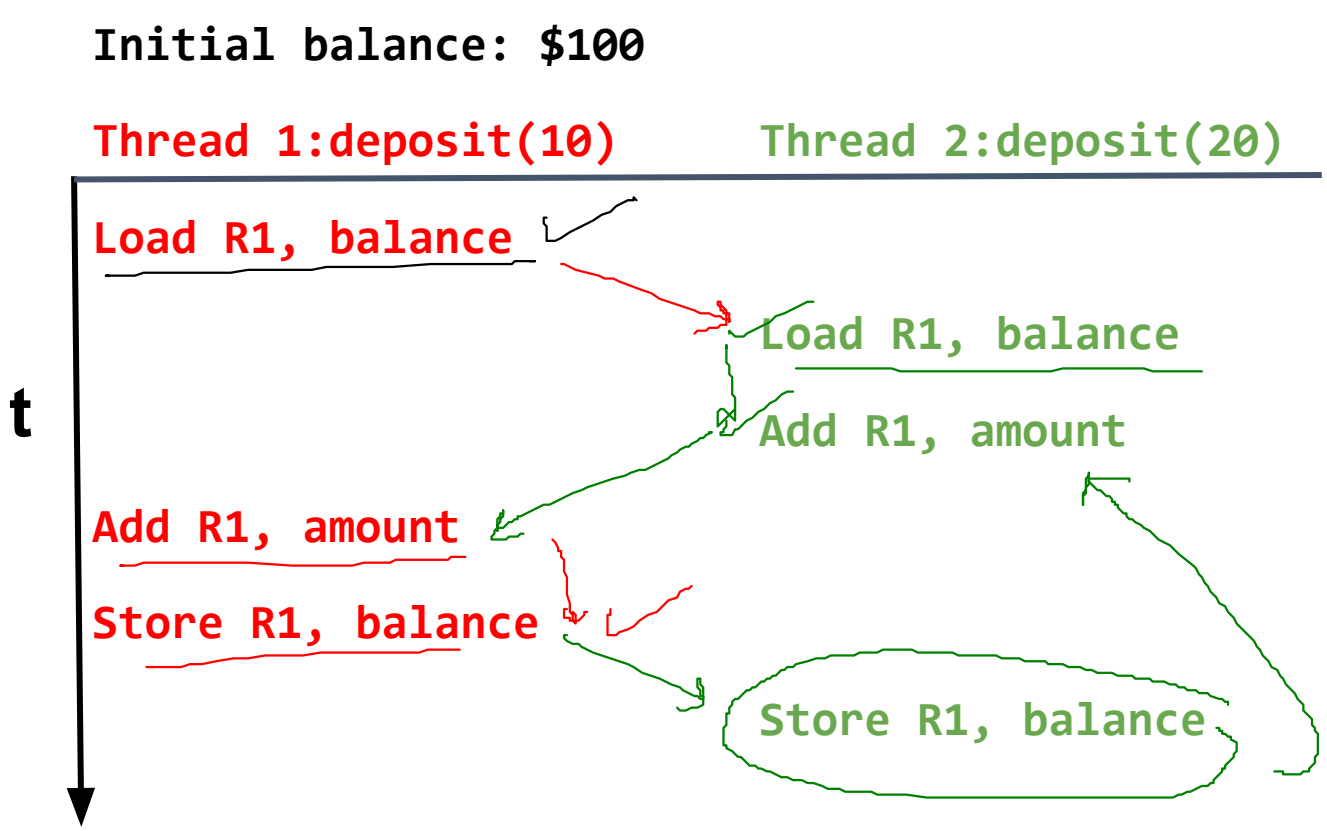


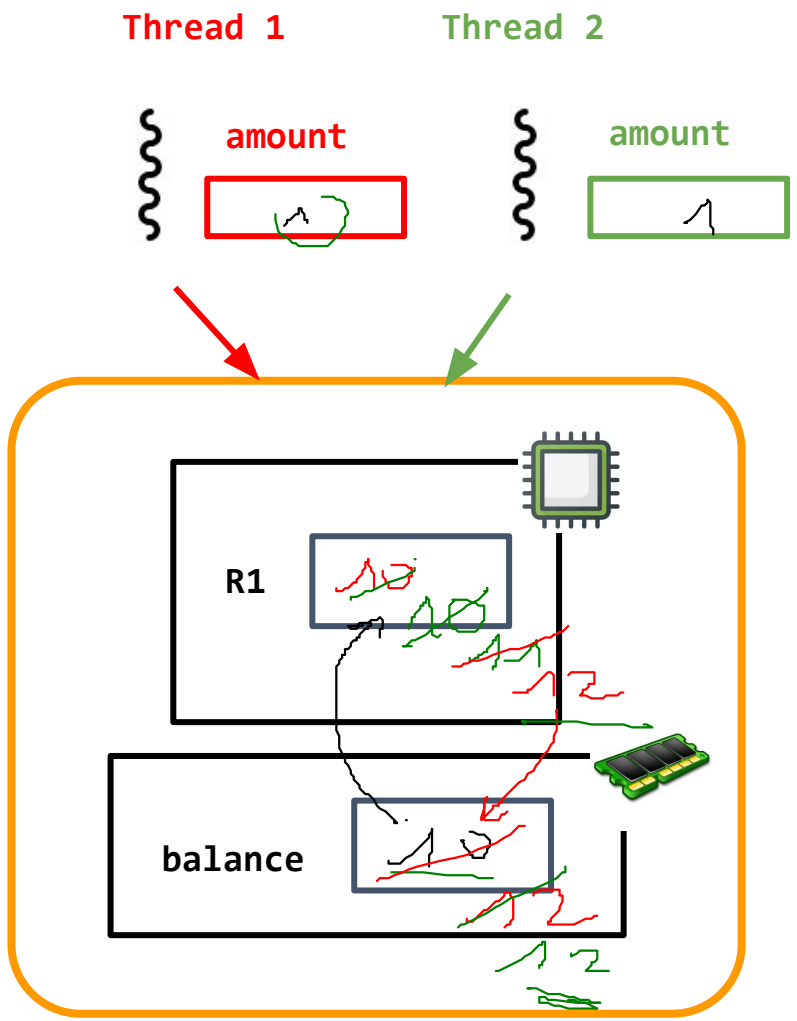
Ejemplo tomado de: http://pages.cs.wisc.edu/~remzi/Classes/537/Fall2005/Lectures/lecture4.pdf

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Recurrencia





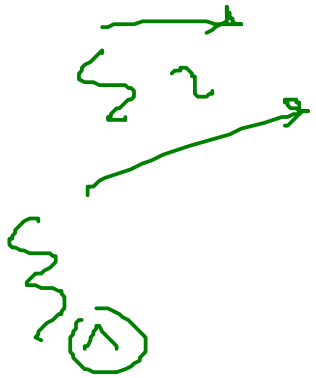
Ejemplo tomado de: http://pages.cs.wisc.edu/~remzi/Classes/537/Fall2005/Lectures/lecture4.pdf

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

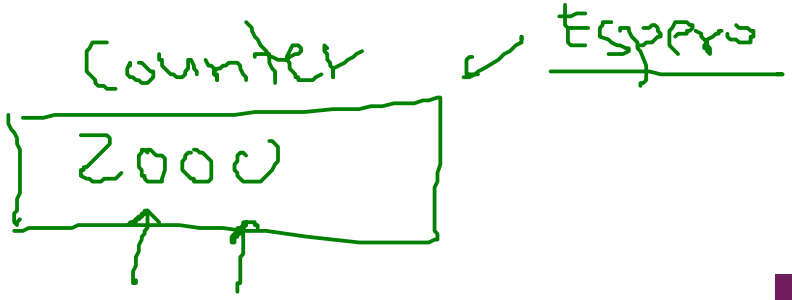
Ejemplo

threads.c



```c
#include <stdio.h>
#include <stdlib.h>
#include "common.h"
#include "common_threads.h"

volatile int counter = 0;
int loops;

void *worker(void *arg) {
    int i;
    for (i = 0; i < loops;
i++) {
     counter++;
    }
    return NULL;
}
```

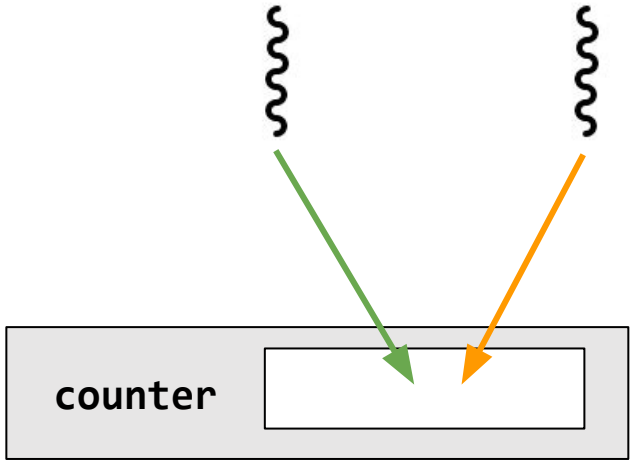
```c
int main(int argc, char *argv[]) {
    if (argc != 2) {
     fprintf(stderr, "usage: threads <loops>\n");
     exit(1);
    }
    loops = atoi(argv[1]);
    pthread_t p1, p2;
    printf("Initial value : %d\n", counter);
    Pthread_create(&p1, NULL, worker, NULL);
    Pthread_create(&p2, NULL, worker, NULL);
    Pthread_join(p1, NULL);
    Pthread_join(p2, NULL);
    printf("Final value   : %d\n", counter);
    return 0;
}
```



Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Ejemplo concurrencia




```
examples — -bash — 84×34
Dannys-MacBook-Pro:examples dannymunera$ ./threads 10
Initial value : 0
Final value : 20
Dannys-MacBook-Pro:examples dannymunera$ ./threads 100
Initial value : 0
Final value : 200
Dannys-MacBook-Pro:examples dannymunera$ ./threads 1000
Initial value : 0
Final value : 2000
Dannys-MacBook-Pro:examples dannymunera$ ./threads 10000
Initial value : 0
Final value : 20000
Dannys-MacBook-Pro:examples dannymunera$ ./threads 100000
Initial value : 0
Final value : 156427
Dannys-MacBook-Pro:examples dannymunera$ ./threads 100000
Initial value : 0
Final value : 163340
Dannys-MacBook-Pro:examples dannymunera$ ./threads 100000
Initial value : 0
Final value : 200000
Dannys-MacBook-Pro:examples dannymunera$ ./threads 100000
Initial value : 0
Final value : 139245
Dannys-MacBook-Pro:examples dannymunera$
```

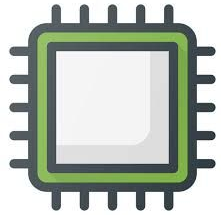
Ver video: C01P11 - C01P12 - Ejemplo concurrencia - ISI485 Sistemas Operativos (link)

Imagen tomada de: Operating Systems Concepts (Silberschatz, Gagne, Galvin)

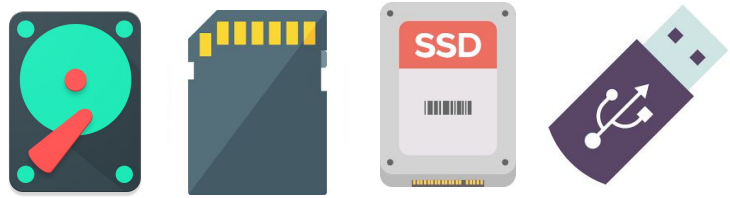
Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Persistencia



- La memoria principal usa una tecnología de **memoria volátil**.
  - Los datos se pierden si deja de haber alimentación.
- Es necesario tener alguna forma de mantener los datos (persistencia).
  - **Hardware**: permite el almacenamiento de datos (discos, duros, discos de estado sólido, etc).



  - **Software**: **sistema de archivos** (controla el disco). Este software es el responsable de almacenar los archivos que crea el usuario de una manera confiable y eficiente en el disco.

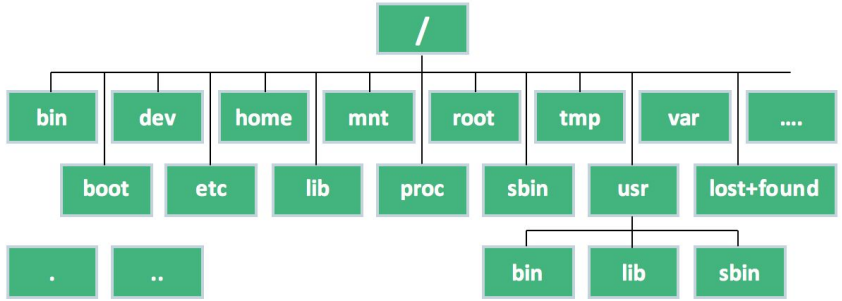


Imagen tomada de: https://cumulusnetworks.com/blog/linux-basics-part-2/

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Ejemplo

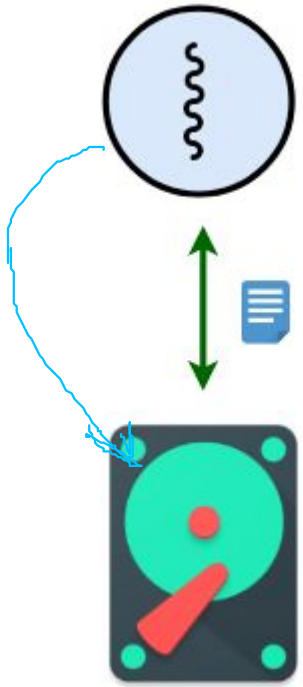
io.c

```c
#include <stdio.h>
#include <unistd.h>
#include <assert.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <string.h>


int main(int argc, char *argv[]) {
    int fd = open("/tmp/file", O_WRONLY | O_CREAT | O_TRUNC, S_IRUSR | S_IWUSR);
    assert(fd >= 0);
    char buffer[20];
    sprintf(buffer, "hello world\n");
    int rc = write(fd, buffer, strlen(buffer));
    assert(rc == (strlen(buffer)));
    fsync(fd);
    close(fd);
    return 0;
}
```

Virtualización de recursos

UNIVERSIDAD DE ANTIOQUIA

Ejemplo persistencia



```
examples — -bash — 84×28
Dannys-MacBook-Pro:examples dannymunera$ ls
1-cpu.c 2-mem   2-mem.c 3-con.c 4-per.c a.out   per     threads
Dannys-MacBook-Pro:examples dannymunera$
Dannys-MacBook-Pro:examples dannymunera$ ./per
Dannys-MacBook-Pro:examples dannymunera$ ls
1-cpu.c         2-mem.c         4-per.c         file_tmp        threads
2-mem           3-con.c         a.out           per
Dannys-MacBook-Pro:examples dannymunera$ cat file_tmp
hello world
Dannys-MacBook-Pro:examples dannymunera$
```

Ver video: C01P13 - Ejemplo Persistencia - ISI485 Sistemas Operativos (link)

Imagen tomada de: Operating Systems Concepts (Silberschatz, Gagne, Galvin)

Agenda

UNIVERSIDAD DE ANTIOQUIA

- Encuesta
- ¿Qué hace un Sistema Operativo?
- Definición de los Sistemas Operativos
- Virtualización de Recursos
- **Operaciones del sistema operativo**
- Objetivos de diseño de los Sistemas Operativos

https://commons.wikimedia.org/wiki/File:NVIDIA_GPU_Accelerator_Block_Diagram.png
https://en.wikipedia.org/wiki/File:GPI_Overview.png

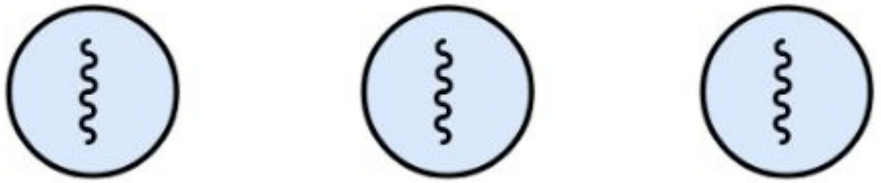
Operaciones del sistema operativo

UNIVERSIDAD DE ANTIOQUIA

Los **modos del procesador** permiten que el sistema operativo se proteja a sí mismo y a otros componentes del sistema.

Modo Usuario

- No privilegiado
- Aplicaciones



Modo Kernel

- Privilegiado
- Kernel del sistema operativo
- Acceso directo al Hardware

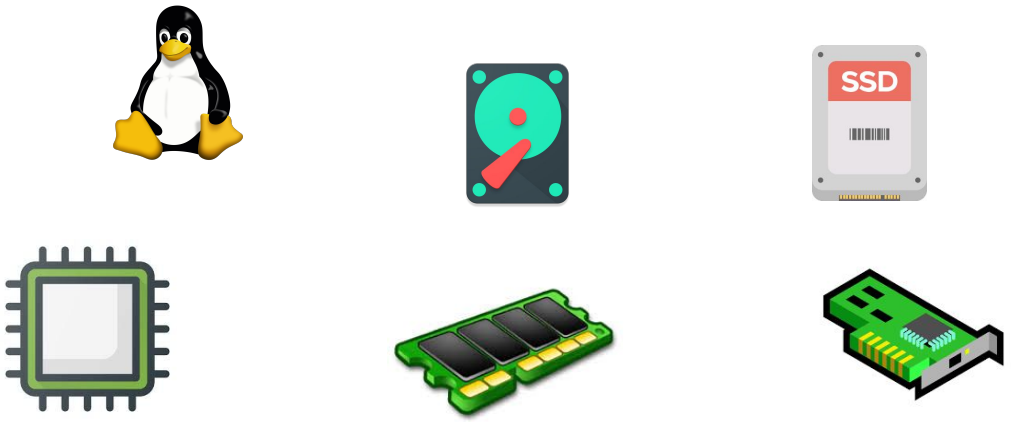


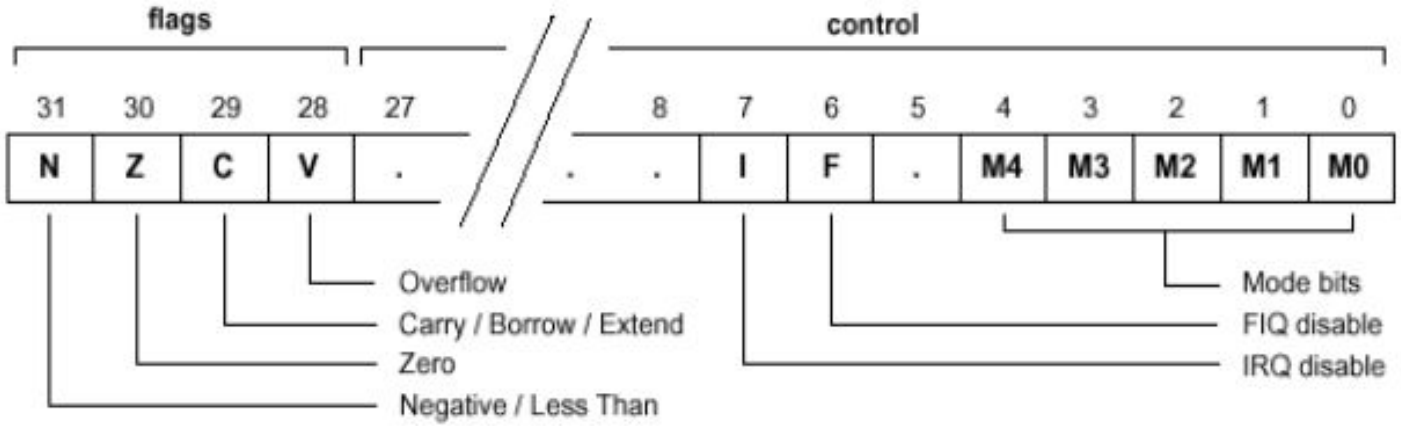
Imagen adaptada de: https://omscs.gatech.edu/cs-6200-introduction-operating-systems

Operaciones del sistema operativo

UNIVERSIDAD DE ANTIOQUIA

El HW proporciona un **bit de modo**

- Diferenciar si el sistema está ejecutando código de usuario o código del kernel.
- Algunas instrucciones son **privilegiadas**, sólo pueden ser ejecutadas en modo kernel.
- Los llamados al sistema cambian la máquina a modo kernel, el retorno de la llamada lo restablece a modo usuario.



Los bits M0, M1, M2, M3 y M4 (M[4:0]) son los bits de modo y determinan el modo en el cual el procesador opera.

Imagen tomada de: http://aelmahmoudy.users.sourceforge.net/electronix/arm/chapter3.htm

Operaciones del sistema operativo

UNIVERSIDAD DE ANTIOQUIA

¿Que pasa cuando una aplicación hace una llamada a sistema?

- El control pasa al SO.
- Se cambia a modo privilegiado.
- El sistema operativo realiza la operación requerida.
- El resultado es retornado a la aplicación.
- El proceso continua en modo usuario.

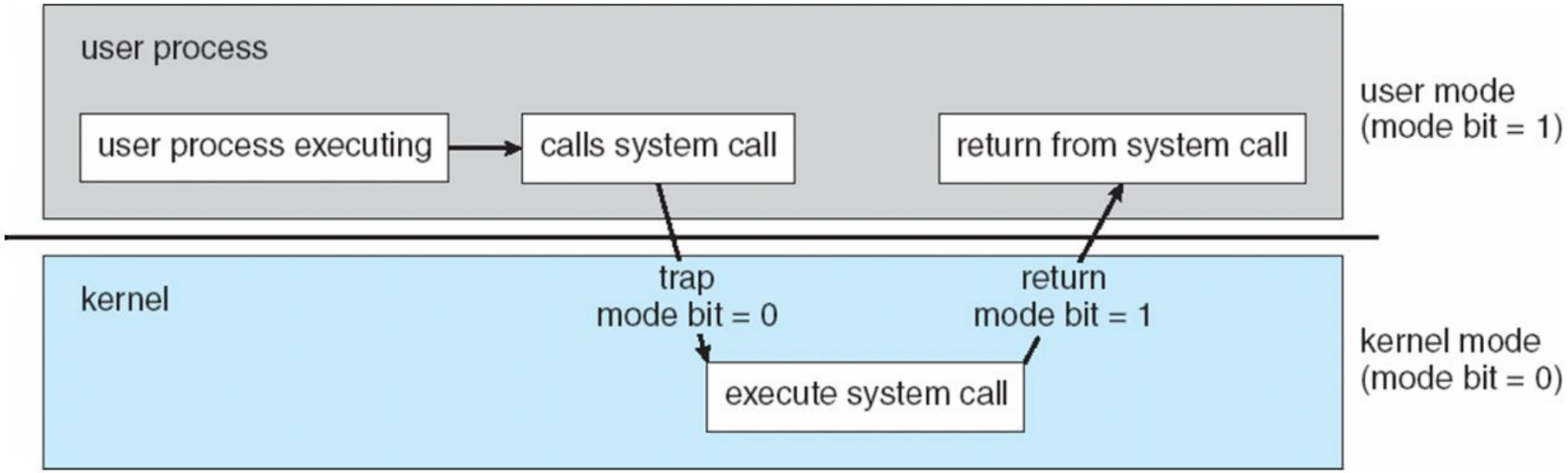


Imagen tomada de: Operating Systems Concepts (Silberschatz, Gagne, Galvin)

Agenda

UNIVERSIDAD DE ANTIOQUIA

- Encuesta
- ¿Qué hace un Sistema Operativo?
- Definición de los Sistemas Operativos
- Virtualización de Recursos
- Operaciones del sistema operativo
- **Objetivos de diseño de los Sistemas Operativos**

Objetivos de diseño de los SO

UNIVERSIDAD DE ANTIOQUIA

1. Construir **abstracciones:**
   - Hacen que el sistema sea **fácil de usar!**
2. Proporcionar un **buen desempeño:**
   - Minimizar el **sobrecosto** (*overhead*) del SO.
   - SO debe proporcionar la virtualización sin generar un sobrecosto excesivo.
3. **Protección** entre aplicaciones:
   - **Aislamiento**: Un mal comportamiento de un proceso **no debe afectar otros** (o al sistema)
4. Confiabilidad:
   - El SO debería ejecutarse **sin problema** por un plazo largo de tiempo.
5. Otros:
   - Eficiencia energética
   - Seguridad
   - Movilidad

Referencias

UNIVERSIDAD DE ANTIOQUIA

- **Operating Systems Three Easy Pieces** (website: Capítulos 1 y 2)
- Videos de youtube: Clase 1 - Introducción a los sistemas operativos (link).
- Página de Remzi H. Arpaci-Dusseau (link)
- Operating System Concepts - Tenth Edition (website)
- Operating Systems Notes (link)

UNIVERSIDAD DE ANTIOQUIA

Curso de sistemas Operativos

Clase 1 - Introducción a los Sistemas Operativos