Sistemas Operativos

David Lopez

Contents

1 conceptos basicos

Un sistema operativo no es mas que un progama.

1.1 Funciones del s.o

Este programa, no obstante, es bastante caracteristico, pues se encarga de **gestionar los recursos hardware**, ofrece a los demas programas una api de **llamadas al sistema** y proporciona **mecanismos de interaccion con el usuario** como el shell.

Es necesario gestionar los recursos hardware por que podemos tener varios programas ejecutandose a la vez, y se hace mas importante con los sistemas multiusuario. Los programas competiran por los recursos, por lo que sera necesario administrarlos correctamente.

Sobre las llamadas al sistema habra algunos ejercicios para familiarizarnos, hay que tener en cuenta, que a diferencia de un curso de desarrollo sobre el kernel de un sistema operativo, aqui nos centraremos en programas de espacio de usuario y nos limitaremos, al menos por el momento a invocar a estas llamadas, no a implementarlas en incluirlas en el s.o , cosa que tambien se puede hacer.

1.2 partes principales del s.o

Podemos distinguir tres partes principales: **nucleo o kernel, servicios e interfaz de usuario**

1.3 clasificacion segun su estructura

- Monolitico (como linux): Tiene todos los componentes integrados en un mismo programa
- Por capas: cada capa ofrece una interfaz a la capa siguiente
- Cliente-Servidor (microkernel): el kernel se reduce a la minima expresion y el resto de utilidades se desarrollan en proceso de usuario

1.4 Como lo usamos ?

Podemos poner el sistema operativo a trabajar, deliberadamente y para que haga la funcion que queremos, escribiendo un programa que realize llamadas al sistema, las interrupciones de los perifericos y los errores tambien provocara la activacion del s.o

1.5 ejemplo de programa que invoca llamadas al sistema

escribir una funcion en C, que actue como el comando cat de unix, es decir, que imprima el contenido del archivo por la salida estandar

```
#include <stdio.h>
#include<fcntl.h>
#include<unistd.h>

#define BUFFSIZE 512

void main(int arg, char *argv[]){

  int fd = 0;
  int cnt = 0;
  char buff[BUFFSIZE];

  fd = open(argv[1], O_RDONLY);

  while((cnt = read(fd, buff, BUFFSIZE)) > 0){
    write(1, buff, cnt);
  }

  close(fd);

```

```
21
22
23
24 }
```

En este codigo, las funciones *open, read, write y close* son llamadas al sistema, cuyo man se puede encontrar en: manual de llamadas al sistema, es fundamental saber utilizar esta herramienta y manejar con solvencia las llamadas al sistema mas habituales.

2 Procesos

2.1 Conceptos basicos

Un proceso es la unidad de procesamiento gestionada por el sistema operativo.

El sistema operativo mantiene una tabla de procesos. En esta tabla se almacena un **bloque de control de proceso** por cada proceso. El BCP contiene la siguiente informacion:

- Identifiacion: PID, USER y relaciones padre-hijo
- Estado del procesador
- Informacion de control del proceso
- Memoria asignada
- Recursos asignados