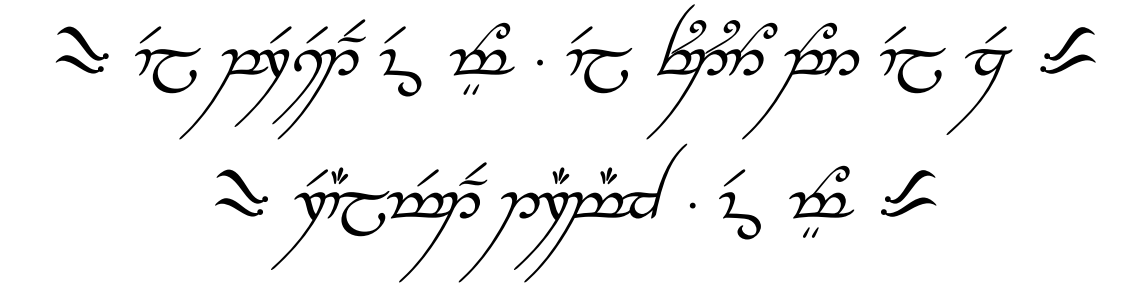
Escuela Técnica Superior de Ingeniería Informática

Asignatura:
Sistemas Operativos

Autor:
Fernando José Mateos Gómez

Ultima Modificacion: **26 de Septiembre del 2021**



Indice

1. Tema 1: Fundamentos

1.1. Sistema Operativo

El *Sistema Operativo* es un software que facilita el uso del sistema informático, "SI", enlazando al usuario con el hardware, mediante las **interfaces entre las aplicaciones y el sistema** y las **interfaces entre el usuario y el sistema**.

1.1.1. Kernel

El kernel o también llamado "núcleo del sistema", es una capa situada "por encima", un nivel de abstracción intermedio entre el *hardware y las herramientas* que hacen la función de *interfaces*, estas herramientas son de 3 tipos:

- **Herramientas del Sistema**: Intérprete de comandos, administración del sistema, CLI ...
- **Herramientas de Desarrollo**: Compiladores, depuradores, IDE's ...
- **Aplicaciones**: Navegador, juegos ...

Linux no es un SO, sino un Kernel

1.2. Procesos e Hilos

Los *procesos* son objetos del "SO" que encapsulan un número determinado de *hilos* en un espacio de memoria, y cuenta con recursos de **hardware y sofware**.
Gracias al *kernel*, creamos una capa de abstracción, por la cual definimos un espacio de memoria propio para cada proceso, que no puede ser usado o visto por otro proceso, ya que cada espacio de memoria está definido para un solo proceso.

Es decir, **los procesos están incomunicados entre si**, sin embargo la cooperación entre procesos se debe al propio kernel, que proporciona herramientas, y además un proceso es capaz de crear otros procesos y acceder al sistema, unicamente solicitando servicios.

En Windows, el espacio estandar que se reserva por cada proceso es de 2Gi

1.2.1. Multiprogramación

Considerando que tenemos varios procesos cargados, y que alguno está bloqueando al sistema (*requiere una cantidad de tiempo muy superior al tiempo que puede cargar instrucciones el procesador*), podemos aprovechar la existencia de varios procesadores y ejecutar varios de estos procesos en cada procesador, aumentando la velocidad.

Podemos suponer que cada procesador ejecutará sus procesos de forma independiente al resto, sin embargo en la realidad, usaremos un **planificador**, *que en vez de planificar procesos, planifica hilos*, que liberará un procesador para que otro "trabaje", para esto el planificador usará algoritmos de planificación tales como el *FIFO*, *LIFO* ...
Estos algoritmos, trabajan siempre bajo la misma premisa:

1. El proceso está **preparado**, este está buscando un procesador para poder ejecutarse y se mantendrá en esta etapa mientras el planificador lo indique.
2. Cuando encuentra un procesador, el proceso se pone en **activo**, y se ejecuta.

3. Al terminar el proceso, se queda **bloqueado**, deja de consumir tiempo del procesador y se mantendrá así hasta que se requiera volver a usarlo y volverá a estar **preparado**

4. De forma intermedia, podemos **apropiarnos** del procesador en medio de la ejecución del proceso, lo que lo llevará a estar **preparado**

El tiempo que tarda un proceso, no tiene relación con el tiempo que consume del procesador, por lo que las operaciones que dependan del sistema, se harán a través de herramientas del sistema.

1.2.2. Hilos

Denominamos a los *hilos* como secuencias de procesos que comparten el mismo espacio de memoria.

Estos nacen debido a la *paralelización* de procesos, cuando tenemos varios procesadores podemos intentar ejecutar varios procesos a la vez, sin embargo si estos procesos dependen del estado de los otros, al no poder compartir recursos entre ellos, usamos un espacio de memoria reservado para un conjunto de procesos, de forma que ahora pueden comunicarse entre si. Solo no es rentable cuando se ejecutan operaciones de cálculo.

1.3. Conceptos básicos sobre Hardware

1.3.1. Organización básica de un Ordenador

El *ordenador* es un conjunto de procesadores, unidos a una memoria y a diversos controladores, que a su vez están conectados a un bus de datos.

Cada procesador accede a su parte de la memoria, **no pueden leerse y escribir a la vez**, pero al resto de la memoria que no está asignada a los procesadores, se podrá acceder de esta forma simultaneamente.

1.3.2. Interrupciones

Es una alteración en la secuencia de ejecución de las instrucciones del procesador. Tiene distintas causas:

- **Interrupción del *Hardware***: Como desconectar algún controlador.
- **Excepciones**
- **Ejecución de instrucciones de petición de interrupción**: Como *INT* o *TRAP*

Las *interrupciones* las tratamos de la siguiente forma:

1. Termina de ejecutar la CPU la instrucción actual.
2. Almacenamos el estado del procesador en un registro.
3. La CPU pasa a modo *supervisor*
4. Determinamos la dirección de una *Subrutina de Servicios de Interrupción*, "SSI", que es una tabla de vectores que almacena codigos de interrupciones.
5. Retornamos el valor de la tabla de SSI y restauramos el estado anterior.
6. Continuamos con la ejecución del proceso.

1.3.3. Excepciones

No son más que errores que suceden durante la ejecución de una instrucción.

1.3.4. Modos de Ejecución de un procesador

El procesador ejecutará una serie de *instrucciones*, operaciones aritméticas; lógicas o de movimiento de datos.

Podemos diferencias dos tipos de modo de ejecución:

- **Supervisor**: Con este modo, el procesador puede ejecutar cualquier instrucción. Para pasar de *supervisor* a *usuario*, no es necesaria cumplir ninguna condición.
- **Usuario**: Solo puede ejecutar las *instrucciones no privilegiadas*, aquellas que no accedan a recursos del *hardware* ni recursos del *software* externos al actual. Para acceder al modo *supervisor*, es necesaria la ejecución de una interrupción, y esto ocurrirá si y solo si, cuando el proceso que quiera ejecutar pertenezca al *kernel*.

2. Tema 2: Sistemas de Archivos

3. Tema 3: Virtualización

4. Tema 4: Contenedores