Introduccion a los Sistemas Operativos

Teoria 1 – 20/8/24:

Sistema Operativo:

* Es un software que esta pensado para administrar otro software.
* Requiere un procesador y memoria
* El SO oculta el HW y presenta a los programas abstracciones mas simples de manejar.
* Usuario 🡪 SO 🡨 Hardware
* Administra los recursos de HW de uno o mas procesos.
* Ejecución simultanea de procesos.

Interrupciones:

* Por Hardware: Son asincrónicas
* Por Software: Se causan por algo que se esta haciendo mal. Por ejemlo: Acceso a memoria indebido, división por 0, buffer overflow. Se causa por el ciclo de execute.

Teoría - 27/8/24:

System Calls:

* Es la forma en que los programas de usuario acceden a los servicios del SO
* Se ejecutan en modo kernel o supervisor
* Los parámetros asociados a las llamadas pueden pasarse de carias maneras: por registros, bloques o tablas en memoria o la pila.
* Ej: count=read(file, buffer, nbytes);

Los programas no funcionan en los dos con el mismo código ya que las System Calls en cada sistema operativo es diferente.

Los compiladores no son los mismos entre sistemas operativos. Entre 32 y 64 bits es diferente, RM e INTEL también son diferentes, etc.

* El codigo: read(file, buffer, nbytes); se traduce en:
  + Push file
  + Push buffer
  + Push nbytes
  + Call

Se usa como una “api” para llegar del mundo usuario al mundo kernel, que hace:

Se busca forzar una interrupción

Para buscar que llamada al sistema busco de todas las que hay

En la llamada a sistema hay un vector de direcciones de funciones

Una vez que estoy en el vector de llamadas al sistema, el sys call handler, es el que sabe a donde avanzar y ejecutando la llamada a sistema.

SYSCALL:

Para programar el clásico “hello world” se necesitan mínimo 2 llamadas al sistema:

* SYSCALL WRITE
* SYSCALL EXIT

El comando man es para ayuda, “man write” en consola te dice que hace el write.

En arquitectura de 32 bits, Linux tiene 3 archivos standard:

* Entrada standard pos 0
* Salida standard pos 1
* Error standard pos 2

En arquitectura de 64 bits

Proceso:

* Es un programa en ejecución
* Para nosotros serán sinónimos: tarea, job y proceso.

Diferencia entre programa y proceso:

Programa:

* Es estático
* No tiene program counter
* Existe desde que se edita hasta que se borra

Proceso:

* Es dinámico
* Tiene program counter
* Su ciclo de vida comprende desde que se solicita ejecutar hasta que termina.

El programa es el “.exe”, cuando vos ejecutas el programa, se crea un proceso que ejecuta el programa. Lo que se ejecuta es el proceso que ejecuta el programa.

El programa nunca cambia, lo que cambia es el proceso ya que cada vez que se abre se crea un proceso que ejecuta este programa, y cuando se cierra se termina la ejecución del proceso.

Los procesos pueden coexistir en un SO, pero si solo tenemos una CPU, se puede ejecutar 1 por vez

Atributos de un proceso:

* Identificación del proceso y del proceso padre.
* Identificación del susuario que lo “disparo”
* Si hay estructura de gruspo, grupo que lo disparo.
* En ambientes multiusuario, desde que terminal y quien lo ejecuta.

Process control block (PCB):

* Estructura de datos asociada al proceso (abstracción)
* Existe 1 por proceso
* Es lo primero que se crea cuando se crea un proceso y lo ultimo que se borra cuando termina.
* Contiene la información asociada con cada proceso:
  + PID, PPID, etc.
  + Valores de los registros de la CPU (PC, AC, etc.)
  + Planificación (estado, prioridad, tiempo consumido, etc.)
  + Ubicación (representación) en memoria.
  + Accouting
  + Entrada salida (estado, pendientes, etc.)