Sistemas Operativos

A nivel de hardware el CPU tiene 3 conjuntos de instrucciones: usuario, privilegiadas y virtualización.

Anteriormente los usuarios podían dar instrucciones privilegiadas que pueden provocar dañar la compu. Ahora los usuarios no tienen acceso a las instrucciones privilegiadas (instrucción que realice modificación de hardware) solo el SO. Que recibe las solicitudes a traves de un sistema bien definido para hacer los cambios.

Cuando se crea un programa en la RAM se le asigna un espacio de direcciones. CPU y SO monitorean si una operación de memoria tiene una dirección valida, es decir que trate de acceder a direcciones de memoria que solo sean las que tiene asignadas el programa. Es por protección para que un código malicioso no inyecte código en un proceso diferente.

El usuario tampoco tiene acceso a las instrucciones de virtualización, que incluso están mas protegidas que las privilegiadas.

07/03

Programa Fork

#nclude <stdio.h>

#nclude <stdlib.h>

#nclude <unistd.h>

#nclude <stdio.h>

Int main(){

pid\_t pid;

pid=fork();

if(pid<0){

printf(“Error”)

}

if(pid==0){ //hijo

printf(“PID: %ld\n”, (long)getpid());

printf(“PPID: %ld\n”, (long)getppid());

}

Else{ //padre

printf(“PID: %ld\n”, (long)getpid());

printf(“PPID: %ld\n”, (long)getppid());

}

Sleep(20);

Return 0;

}

/////////////////////

Int main(){

pid\_t pid;

pid=fork();

if(pid<0){

printf(“Error”);

}

if(pid==0){ //hijo

exepid

}

Else{ //padre

printf(“PID: %ld\n”, (long)getpid());

printf(“PPID: %ld\n”, (long)getppid());

}

Sleep(20);

Return 0;

}

* Una vez que se llama exec el hijo pierde todo lo que tenia en el espacio de direcciones

//NOTAS

Proceso zombi: proceso que ya termino la ejecución pero como el padre termino antes las tablas quedaron con referencias al proceso que ya termino. // proceso cuyos hijos no enviaron señal de terminación adecuada y deja referencias a procesos que no existen. -> consecuencia de que el padre termino antes que los hijos.

Manual Linux

Lo primero que hay que hacer cuando no sabe algo es leer el manual…

ARBOL DE PROCESOS

Commando: python3

A forks b

A hace fork de b

Para verlo poco a poco utilice la bandera -e

-s es la semilla

QUIZ abierto de hoy al jueves

Tarea para la próxima semana