

## Laboratorio 5

- Funcionamiento y sintaxis de uso de structs.

Con esto se puede combinar data de diferentes tipos, el uso de structs es un tipo de dato definido por el usuario, son utilizados para representar un registro funcionan de manera muy similar a un class

```
Struct [structure tag]{
```

```
‘
‘
‘
```

```
}[one or more structure variables];
```

- Propósito y directivas del preprocesador.

el preprocesador se utiliza para que los programas de origen sean mas faciles de cambiar y compilar, las directivas del archivo de codigo indican acciones especificas.

Directivas:

```
#if
```

```
#else
```

```
#line
```

```
#import
```

```
#using
```

```
#undef
```

- Diferencia entre \*y &en el manejo de referencias a memoria (punteros).

& nos da la dirección de memoria de una variable

\* Es una variable que guarda la dirección de la memoria de otra variable

- Propósito y modo de uso de APT y dpkg.

Hablando de APT esto gestiona paquetes esta maneja instalaciones de paquetes de paquetes Linux

Con respecto al comando dpkg este instala un paquete específico y notificará de cualquier dependencia

- ¿Cuál es el propósito de los archivos sched.h modificados?

Crear estructuras constantes que ayudan a implementar la política de calendarización

- ¿Cuál es el propósito de la definición incluida y las definiciones existentes en el archivo?

El propósito de estas definiciones es utilizar constantes para usar la política de calendarización a ser implementada por el sistema

- ¿Qué es una tasken Linux?

Como su nombre lo dice es un proceso, un proceso en linux

- ¿Cuál es el propósito de task\_structy cuál es su análogo en Windows?

Su principal proposito es la implementacion de PCB en linux se conoce como KPROCESS. mas que todo es una estructura de datos del kernel de linux

•¿Qué información contiene sched\_param?

Cuando se almacena se puede guardar un identificador y un cuantificador de tiempo. Eos sched\_param almacena parámetros que utilizan las políticas de calendarización

•¿Para qué sirve la función rt\_policyy para qué sirve la llamada unlikelyen ella?

Como nos dice el nombre de rt\_plicyy y de unlikely, estos no dan la capacidad de saber si los datos que pasan son probables tambien ayuda con la velocidad de ejecución. rt\_policy se utiliza en la calendarización mas que todo.

•¿Qué tipo de tareas calendariza la políticaEDF, en vista del método modificado?

para deadlines, estas se saben que son deadlines por los cambios realizados y el nombre de la póliza.

•Describala precedencia de prioridades para las políticas EDF, RT y CFS, de acuerdo con los cambios realizados hasta ahora.

EDF-RT-CFS Mas alto es EDF, bajo CFS

•Explique el contenido de la estructura casio\_task.

Esta tiene una tarea llamada sched casio policy. este tiene un nodo en el cual se usa para ejecutar la calendarización de casio \_task. también esta tiene una lista doblemente encadenada. Además muestra una referencia a un proceso y campo numérico.

•Explique el propósito y contenido de la estructura casio\_rq.

Con respecto a su contenido tiene un procesador que mantiene una cola de procesos esperando a ejecutarse. esto es para asegurarse que los procesos estén en ejecución.

•¿Qué es y para qué sirve el tipo atomic\_t? Describa brevemente los conceptos de operaciones RMW (read-modify-write) y mapeode dispositivos en memoria (MMIO).

Bueno para esto hayque esplicarlo en orden:

- atomic\_t variable de tipo entera que tiene sin necesidad de bloqueo explicito un grupo de operacion
- RMW se puede decir de este que es como un grupo de operadores atómicos y tienen valores nuevos.
- MMIO esta es explicita mente un método de direccionamiento de memoria esto puede construir el sistema de manera que el CPU registra ram

•¿Qué indica el campo .nextde esta estructura?

,next muestra el siguiente case de calendarización

•Tomando en cuenta las funciones para manejo de lista y red-black treede casio\_tasks, explique el ciclo de vida de una casio\_taskdesde el momento en el que se le asigna esta clase de calendarización mediante sched\_setscheduler. El objetivo es que indique el orden y los escenarios en los que se ejecutan estas funciones, así como las estructuras de datos por las que pasa. ¿Por quéseguardan las casio\_tasksen un red-black treey en una lista encadenada?

Esto lleva varios pasos, vamos a enlistarlos:

- Se comienza asignando la clase de calendarizacion, des pues se agrega un proceso a la lista
- Un casio\_task cambia su estado a ready se agrega red-black-tree

- se modifica su deadline absoluta
- Para salir del estado ready se debe de sacar RBT y se termina la ejecucion
- Una lista encadenada y RBT son encargadas para la ejecucion u almacenar casio\_tasks.

•¿Cuándo preemptea una casio\_task a la task actualmente en ejecución?

Estos tienen una prioridad alta solo pasa cuando el proceso que se corre no es casio task, esto solo puede suceder cuando casio\_task esta a la izquierda de RBT que es menor a la tarea en ejecución.

•Ejecute nuevamente el archivo casio\_systemtal como se hizo al inicio del laboratorio, pero guardando los resultados en un archivo diferente. Adjunte ambos archivos de resultados de casio\_system a su entrega, comentando sobre sus diferencias.

- pre\_casio.txt solamente tiene un proceso
- post\_casio.txt Este tuvo mas procesos, prioridad alta, sus tiempo de ejecucion se tardaron mas en su ejecucion

Las similitudes son que cuando termina un proceso SIGUSR1, despues SIGUSR2 todos los task fueron completados al final.

•Ubique el archivo de log de eventos registrados por la calendarización implementada. Adjunte este archivo con su entrega.

Archivo

•Agregue comentarios explicativos a los archivos casio\_task.c y casio\_system.c que permitan entender el propósito y funcionamiento de este código. Asegúrese de aclarar el uso de instrucciones y estructuras que no conozca (como, por ejemplo, los timers y la estructura itimerval). ¿Qué información contiene el archivo system que se especifica como argumento en la ejecución de casio\_system?

La manera en que se va a probar el calendarizador estas tienen informacion de configuracion. Bueno estos se pueden enlistar por:

Retrasos minimos y maximos, numero de proceso, parametros en segundos, tareas subsecuentes.

•Investigue el concepto de aislamiento temporal en relación a procesos. Explique cómo el calendarizador SCHED\_DEADLINE, introducido en la versión 3.14 del kernel de Linux, añade el algoritmo EDF para lograr aislamiento temporal.

- Muy bien, cuando hablamos de sistemas operativos cuando tenemos el aislamiento temporal esta es una capacidad que garantiza que un proceso es independiente de otros procesos con lo cual no esta afectado por otros
- SCHED\_DEADLINE estos funcionan tan bien que se puede dividir en varias etapas, estos procesos pasan por deadline un periodo de realización y un requisito. Entonces el sistema operativo hace prueba verifica que los procesos esta correctos y también verifica tiempo.