Universidad del Valle de Guatemala0.

Sistemas Operativos

Carlos Humberto Chew Roldán

17507

LABORATORIO 4

1. Funcionamiento y sintaxis de uso struct: esta es una declaración de datos, que esta adentro de un bloque de memoria. Lo que facilita trabajar con un puntero. Esto hace referencia a un bloque continuo de la memoria física.

Struct nombre {

Tipo variableA;

Tipo variableB;

}

1. Propósito y directivas del preprocesador

La tarea de esto es procesar las entradas del programa utilizando los headers #include #define #if.

1. Diferencia entre \* y & en el manejo de referencia a memoria

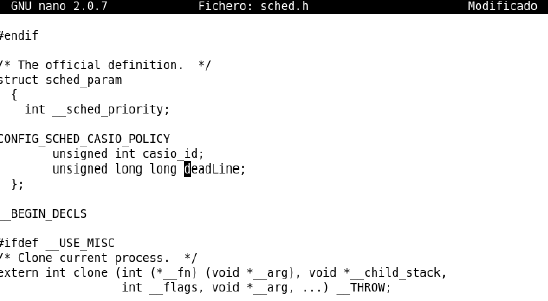
(\*) se identifica como el operador de referencia y retorna un puntero con la dirección de memoria.

(&) se identifica como el puntero que retorna la dirección de memoria en donde se almacena el valor.

1. Propósito y modo de uso de APT y dpkg

El APT tiene como propósito el manejo de software, que sirve para la instalación o eliminación de programas. Sus modos son; update, upgrade y gull-upgrade. El dpkg es una herramienta para instalar, compilar y remover archivos.

1. ¿Cuál es el propósito de los archivos sched\_h modificados?

Esto contiene los parámetros de calendarización para implementar políticas ya establecidas.

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

Este incluye definiciones que son para las políticas de calendarización con mayor queue de prioridad. La nueva definición son las nuevas políticas y rango de prioridad.

1. ¿Qué es un task en Linux?

Son procesos que deben ser realizados por el kernel. O entran a la calendarización.

1. ¿Cuál es el propósito de task\_struct y cual es si análogo en Windows?

Esto describe el proceso, en Windows se llama EPROCESS.

1. ¿Qué información contiene sched\_param?

Esto incluye los parámetros de programación del proceso.

1. ¿Para qué sirve la función rt\_policy para qué sirve la llamada unlikely en ella?

Esta función no indica si las políticas de calendarización pertenecen a proceso en tiempo real. Unlikely es el optimizador del compilador.

1. Explique el contenido de la estructura casio\_task.

* Task\_struct es la información de la tarea
* Casio\_list\_node es la posición de la tarea en la ready queue
* Absolute\_deadline es el deadline de la tarea
* Asio\_rb es el nodo del árbol red-black en el que se encuentra la tarea

1. Explique el propósito de la estructura casio\_rq

Es la información de la ready queue que se encuentra dentro de casio.

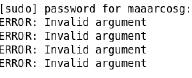
1. ¿Qué indica el campo next de esta estructura?

Es la siguiente tarea que se va a ejecutar.

1. ¿Por qué se guardan las casio\_tasks en un red-black tree y en una lista en cadenada?

Porque balancea la cantidad de tareas.

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

Cuando el task está listo para ser ejecutado.