

Universidad del Valle de Guatemala
Sistemas Operativos
Tomás Gálvez



Laboratorio 4

Jorge Eduardo Súchite
Carnet 15293
Agosto 26, de 2018

Inciso J:

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

El propósito de los sched.h modificado es la implementación del algoritmo de EDF primer tiempo real.

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

Se está tratando de definir las banderas y qué hacen cada una de ellas, al igual que la salida.

La definición que le agregamos trata de poner la definición del Scheduler de casio y asignarle un valor a dicha bandera.

Inciso K:

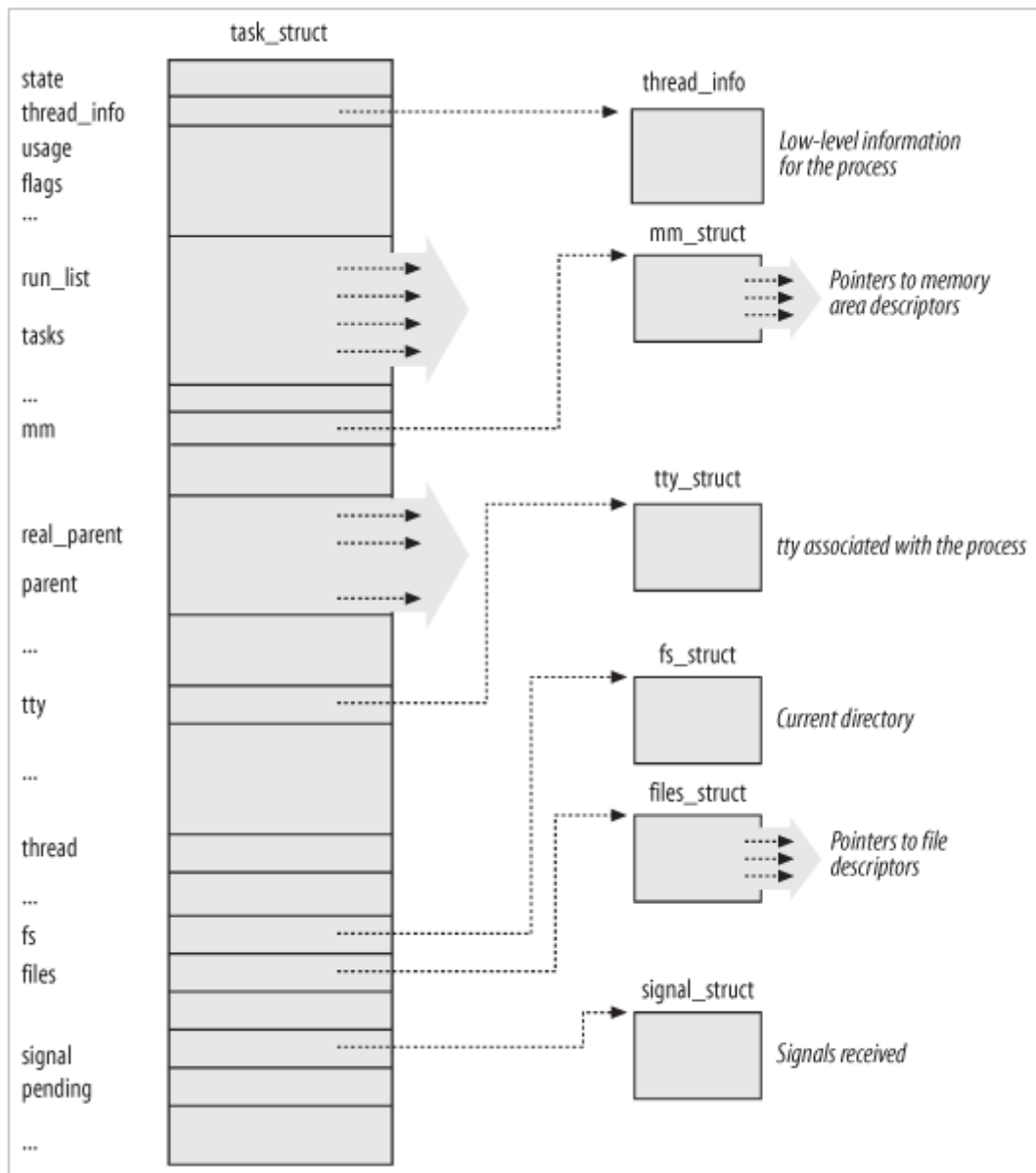
- **¿Qué es una task en Linux?**

Se le puede considerar un proceso o tarea puesto que cada una tiene derechos y responsabilidades a llevar a cabo en el sistema operativo de linux .

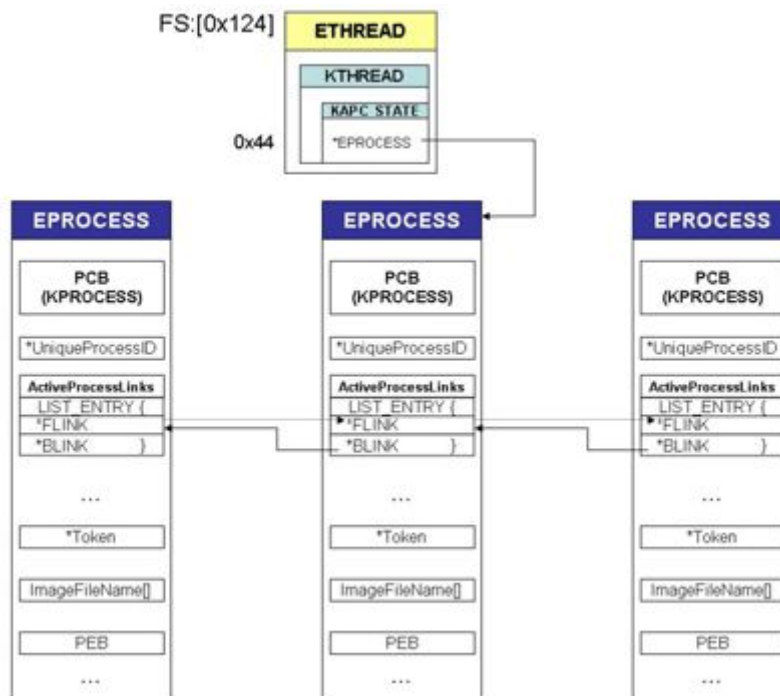
Se le puede llamar como dinámico puesto que, cambia constantemente en el tiempo a medida que se va ejecutando sus instrucciones del código máquina. En ella pueden encontrarse, además de todas las instrucciones a llevarse a cabo, también los parámetros de rutina; direcciones y variables guardadas. Un task es independiente y si este alla, no causará que otra task lo haga también.

- **¿Cuál es el propósito de task_struct y cuál es su análogo en Windows?**

El task_struct es utilizado por linux para gestionar,describir y organizar procesos y cada proceso tiene que tener una estructura de datos ("task_struct")



Su análogo en el Sistema Operativo Windows es llamado **EPROCESS** con lo cual windows identifica un proceso para realizar dicho proceso.



Inciso I:

- ¿Qué información contiene sched_param?

Contiene los parámetros establecidos para la programación de la política de programación para los procesos o subprocesos identificados.

sched_priority está contenida en sched_param y su funcionalidad se activa cuando ya se obtienen los parámetros de programación este amiguito pinte la prioridad que se le asignó a la secuencia de dicho proceso.

sched_curpriority al momento también que se obtienen los parámetros de programación se fija en la prioridad que se le está

dando a un proceso o subprocesso. Asigna un valor para que el núcleo pueda tomar la decisión de agarrarlo.

sched_ss_low_priority muestra el de baja prioridad en el hilo que se está ejecutando.

sched_ss_max_repl muestra la cantidad de veces que se programará un reabastecimiento (en ocasiones por bloqueo) para luego caer a un nivel de baja prioridad.

sched_ss_repl_period contiene el tiempo que se ha de utilizar para programar dicho reabastecimiento de lo establecido para la ejecución después de haber sido bloqueado.

sched_ss_init_budget es el tiempo que debe usarse para el presupuesto de ejecución del hilo. El hilo se ha de ejecutar en el nivel de alta prioridad, su tiempo de ejecución (como ya se mencionó) se extrae del presupuesto. Una vez que el presupuesto se agota el hilo cae al nivel de baja prioridad, donde, si es posible se puede hacer arreglos de prioridad para seguir ejecutándose hasta que se reabastece el presupuesto de ejecución.

Nota: ***sched_priority > sched_ss_low_priority***
sched_ss_max_repl < SS_REPL_MAX
sched_ss_init_budget > sched_ss_repl_period

Inciso m:

- **¿Qué tipo de tareas calendariza la póliza EDF?**

Las tareas que ha de calendarizar tienen que tener especificadas sus deadlines o sean previsibles. Ambientes de operación en los

que los calendarizadores están en un sistema de tiempo real tienen muchas de estas tareas.

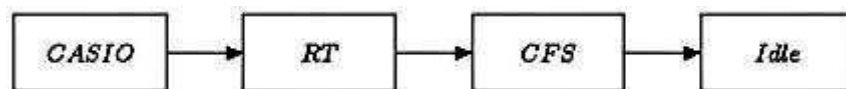
- **¿Para qué sirve la función `rt_policy` y para qué sirve la llamada `unlikely` en ella?**

La función `rt_policy` se usa para decidir si una política de planificación dada pertenece a la clase en tiempo real o no.

`Unlikely`, es macro que ayuda al compilador para decidir si se puede tomar o no la predicción de un proceso que se repite.

Inciso n:

Indique la precedencia de prioridades para las pólizas EDF, RT y CFS, de acuerdo con los cambios realizados hasta ahora.



(imagen obtenida del tutorial)

Sabemos que el CASIO implementa el algoritmo EDF, entonces como prioridad la póliza casio va primero, segundo RT, tercero cfs.

Inciso p:

Explique el contenido de la estructura `casio_task`.

Se conforma por un nodo, un elemento no definido que guarda el deadline de la tarea con un elemento de cabeza de la lista de las tareas y un puntero que apunta a la próxima tarea.

Inciso q:

Explique el propósito de la estructura casio_rq

Guarda la dirección donde está la tarea x y ayuda a ponerla en la cola de las tareas.

Inciso w:

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

Es un puntero hacia sched_class que se utiliza para organizar los módulos de prioridad y linkearlos a la lista. Buscará una tarea que se pueda ejecutar de cada módulo en una prioridad decreciente de orden.

Inciso x:

¿Por qué se guardan las casio_tasks en un red-black tree y en una lista encadenada?

Se inicia una tarea “CASIO” como cualquier tarea en el sistema, utilizando las llamadas de sistema fork o clon.

(queue) Para ponerla en la cola se manda a llamar una tarea y esta ingresa a un estado ejecutable. Recibe los dos punteros; uno para la cola de ejecución del procesador (rq) y otro para la tarea que ingresa un estado ejecutable (p).

Luego de invocar dicha función, se invoca a la función de la lista de tareas “find_casio” para luego así obtener el puntero de la tarea Struc_casio almacenada en la lista vinculada a la de la estructura casio_rq que esta va a apuntar a la tarea p. Para este entonces, se debe de actualizar la fecha límite absoluta e inserta casio_task en el rb_tree.

Al final, el kernel nativo “sched_clock” devuelve la hora actual en anosegundos (definida en /kernel_source_code/kernel/sched.c). Además, registra este evento en el sistema de registros de CASIO.

Cuando una tarea CASIO no se puede ejecutar , se manda a llamar a la función enqueue_task_casio para que deshaga el trabajo de la función queue_task_casio.

check_preempt_curr_casio function verifica si la tarea que se está ejecutando actualmente debe ser la indicada. La cual sirve también para poner en cola o dequearse una tarea para que establezca un indicador que indica al núcleo del planificador de prioridad que la tarea se está ejecutando actualmente, debe tener prioridad.

Bueno, luego de una explicación medio larga.

Las casio_task se guardan en un red_black_tree para que se pueda tomar decisiones sobre las tareas que, ya sea por su deadline o la prioridad que tengan se ha de tomar decisiones para que el procesador ejecute dicha tarea y si está ejecutando una tarea se tiene que decidir si esta es la indicada a la que se le debe dar la prioridad. También, si no se ejecuta una tarea CASIO en el procesador., esta va a devolver null y el núcleo planificador de prioridad va intentar buscar en todo el árbol una tarea que sea la siguiente clase de baja prioridad.

Y una lista encadenada pues, me imagino que es para guardar resultados de los procesos para luego compararlos y ordenarlos para luego mostrar los procesos que se necesiten.

Terminado hasta la aa