

Lab # 5

Funcionamiento y sintaxis de uso de structs.

Para los lenguajes de programación que no son class oriented usan structs que es practicamente definir una variable que contenga informacion de diferente tipo.

Syntaxis

```
Struct [def] {  
    Type varname;  
};
```

Propósito y directivas del preprocesador

Son como anotaciones para poder trabajar en diferentes environments por ejemplo en swift existe `if #available(iosxxx,*)` para poder chequear si esta en una version de ios o si esta en ios en lo absoluto. Algo importante a mencionar es que no todas las directivas son de condiciones existen los imports o pragma que se utiliza para OpenMP

Directivas

- `#define`
- `#elif`
- `#else`
- `#endif`
- `#error`
- `#if`
- `#ifdef`
- `#ifndef`
- `#import`
- `#include`
- `#line`
- `#pragma`
- `#undef`
- `#using`

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

* Es un puntero a una variable.
& nos retorna la dirección en memoria de la variable.

Propósito y modo de uso de APT y dpkg.

Ambos son manejadores de paquetes para linux based O.S. pero dpkg es para .deb packages. Una diferencia entre ellos es que dpkg es una herramienta de bajo nivel mientras apt no lo es en apt se maneja un source list que es la que alimenta

Modo de uso APT
Apt-get install package_name

Modo de uso dpkg
dpkg -i filename.deb

Inciso J

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

Los archivos <sched.h> define la estructura de los sched_params que se contiene para casa sched policy. En general los .h son headers para los .c.

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

En el kernel_dir definimos una nueva policy si encontramos su config
Y en usr/include/bits/sched.h se define con la directiva SCHED_CASIO con un valor igual al del kernel_dir

Inciso k

¿Que es una task en linux?

Se refiere a una unidad de ejecución que puede compartir varios recursos del sistema con otros tasks.

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

Es la estructura que contiene toda la información acerca de un proceso.
Su analogo en windows es TASK_TRIGGER

Inciso l

¿Qué información contiene sched_param?

Tiene la prioridad del scheduled task

Inciso m

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

Es para decidir si un policy de scheduling pertenece a un real time class o no
s used to decide if a given scheduling policy belongs to the real-time class (SCHED_RR and SCHED_FIFO) or not.

Unlikely sirve para optimizar branch prediction

¿Qué tipo de tareas calendariza la política EDF, en vista del método modificado?

Calendariza conforme a los primeros respecto a su deadline

Inciso m

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

EDF

Como anteriormente mencionado sería conforme al primer deadline que es el que estamos implementando.

RT

CFS

Se pasan al queue de acuerdo al peso de cada Task

Inciso p

Explique el contenido de la estructura casio_task.

Rb_node

el el nodo del redblack tree

unsigned long long absolute_deadline

Hace referencia al deadline como estamos usando EDF en nuestro CASIO POLICY

Struct list_head casio_list_node;

Nuestra cabeza en los nodos

Struct task_struct* task;

Pues aca tenemos un puntero a nuestro task structure (anteriormente mencionamos contiene toda la info del task) aca vamos con puntero para siempre apuntar al mismo.

Inciso q

Explique el contenido de la estructura `casio_rq`.

Tiene la referencia de un nodo en un red black tree para calendarizar la ejecución.

Tiene la chained list por las que se registran las `casio_tasks`

Tiene referencia al deadline.

¿Qué es y para qué sirve el tipo `atomic_t`? Describa brevemente los conceptos de operaciones RMW (read-modify-write) y mapeo de dispositivos en memoria (MMIO).

atomic_t

El tipo `atomic_t` es una variable de tipo integer que ya cuenta con un set de operaciones que aseguran que sea atómica sin la necesidad de estar bloqueando explícitamente.

Tiene casos de usos como los threads ya que solo uno podrá alterar el valor a la vez siempre será correcto.

RMW

Es una clase de operaciones atómicas que leen, modifican y escriben un valor en memoria.

MMIO

Es un método de IO entre el CPU y dispositivos periféricos en una computadoras.

MMIO usa el mismo espacio en memoria para referirse tanto a memorias y dispositivos de I/O. La memoria y registros de los dispositivos de I/O son mapeados a direcciones en memoria y cuando una dirección es accesada por el CPU puede referir a una porción de la memoria física de la RAM

Inciso w

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

Hace referencia al siguiente en el queue dependiendo de las prioridades.

Inciso x

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

`casio_tasks`

La razón por la cual se usa un red black tree es por la ventaja que un single top-down se puede usar tanto en inserts cómo deleted.

Ofrece un garantías en el worst case scenario de el insert, del delete y de el algoritmo para búsqueda.

Todo camino del root a una hoja tiene la misma longitud. Cada nodo solo puede tener 2 children.

lista encadenada

Se utiliza una lista encadenada porque como su orden no se determina por su posición física en memoria si no que en lugar de eso cada elemento apunta al otro continuando una secuencia seria una secuencia de nodos que es la que más nos conviene.

¿Cuándo preemptea una `casio_task` a la task actualmente en ejecución?

Cuando el schedule se cumple