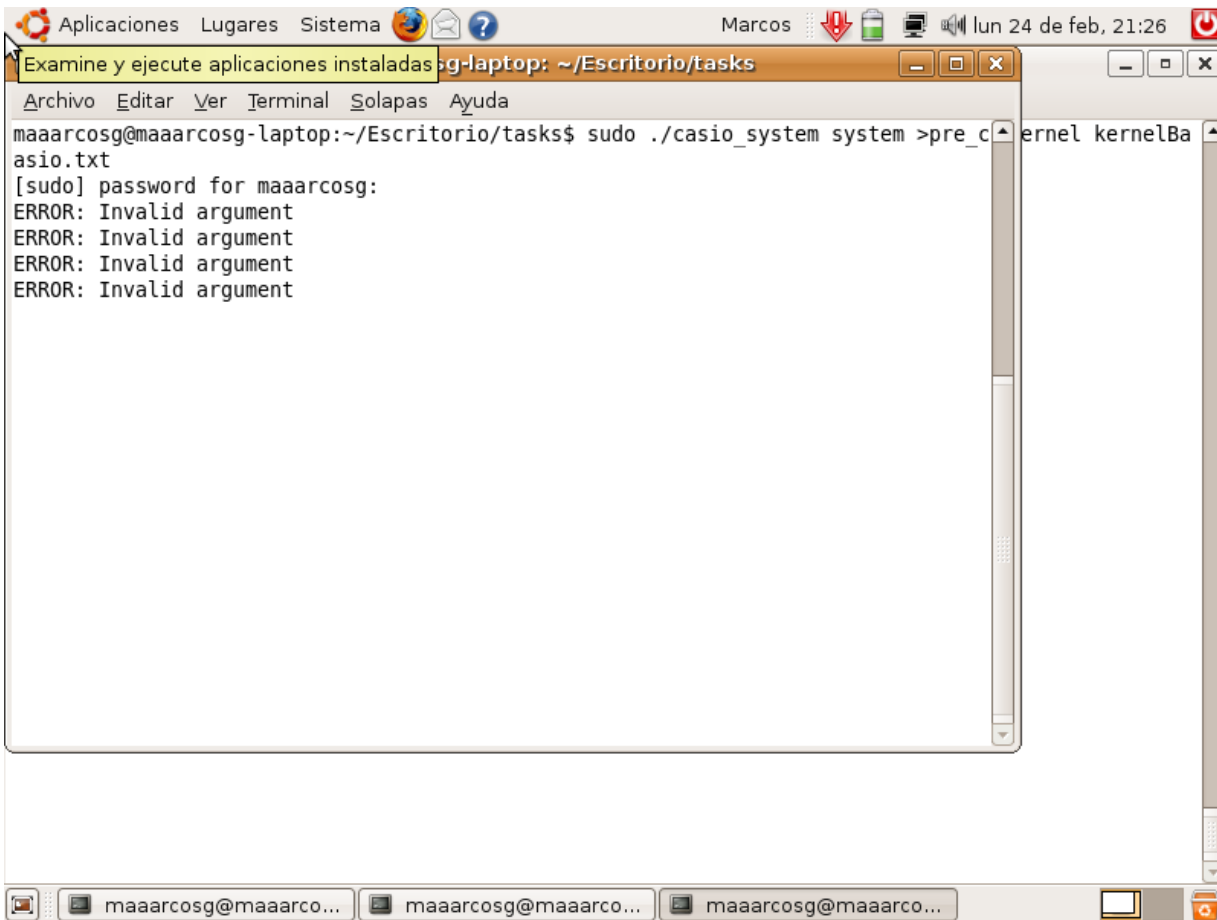


Laboratorio No. 4



```
maaacosg@maaacosg-laptop:~/Escritorio/tasks$ sudo ./casio_system system >pre_c
[sudo] password for maaarcosg:
ERROR: Invalid argument
ERROR: Invalid argument
ERROR: Invalid argument
ERROR: Invalid argument
```

Funcionamiento y sintaxis de uso struct

El funcionamiento de *struct* es una declaración de tipo de datos dentro de un bloque de memoria, lo que nos permite trabajar con variables dentro de un puntero. Además, hace referencia directa a un bloque continuo de la memoria física. La sintaxis es la siguiente:

```
Struct tag_name {
    Type variable1;
    Type variable2;
}
```

Propósito y directivas del preprocesador

El propósito del preprocesador es procesar la entrada del programa utilizando las directivas `#include`, `#define` o incluso `#if`, dichas directivas no son específicas de C.

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

(*) es el operador de referencia, lo cual retorna un puntero con la dirección de memoria.

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

Propósito y modo de uso de APT y dpkg

El APT es un sistema de manejo de software, el cual puede servir para instalación o eliminación de programas, los modos con los cuales se pueden usar son: update, upgrade y full-upgrade. En cambio *dpkg* originario de Debian, es una herramienta para instalar, compilar, remover archivos con el tipo (.deb)

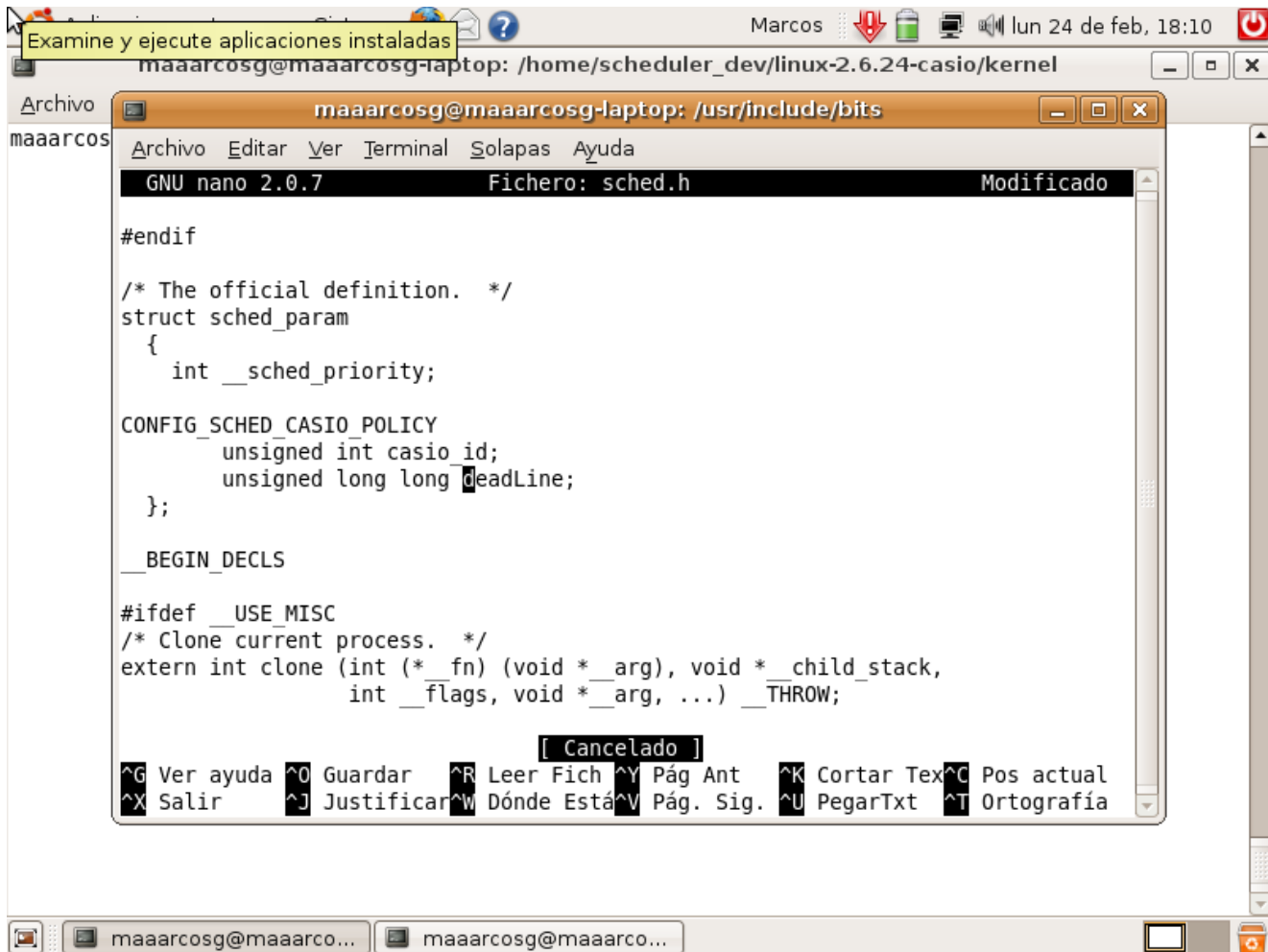


Imagen No. 2 Pruebas de edición de archivo sched.h

- **¿Cuál es el propósito de los archivos sched.h modificados?**
Contiene los parámetros de calendarización para la implementación de las políticas establecidas.
- **¿Cuál es el propósito de la definición incluida y las definiciones existentes en el archivo?**
Las definiciones incluidas son las políticas de calendarización con mayor cola de prioridad. La nueva definición son las nuevas políticas y rango de prioridad.
- **¿Qué es una task en Linux?**
Son todos los procesos que entran a la calendarización del kernel.
- **¿Cuál es el propósito de task_struct y cuál es su análogo en Windows?**
Task_struct describe el proceso, en el sistema operativo Windows este proceso se llama EPROCESS
- **¿Qué información contiene sched_param?**
Los parámetros de programación del proceso.
- **¿Para qué sirve la función rt_policy y para qué sirve la llamada unlikely en ella?**
Rt_policy es la función que no indica si las políticas de calendarización pertenecen a proceso en tiempo real. Unlikely optimizador del compilador.

- **Explique el contenido de la estructura `casio_task`.**

Son los procesos que se encuentran en CASIO de mayor prioridad.

- ✓ `asio_rb_node` es el nodo del árbol red-black en el que se encuentra la tarea.
- ✓ `absolute_deadline` es el deadline de la tarea.
- ✓ `casio_list_node` es la posición de la tarea en el *ready queue*
- ✓ `task_struct` es la información de la tarea.

- **Explique el propósito de la estructura `casio_rq`**

Es la información del *ready queue* política que se encuentra dentro de CASIO.

- **¿Qué indica el campo `next` de esta estructura?**

Es la siguiente tarea o hilo que se va a ejecutar.

- ¿Por qué se guardan las `casio_tasks` en un red-black tree y en una lista en cadenaada?
Porque cuando es un proceso a
- ¿Cuándo *preemptea* una `casio_task` a la task actualmente en ejecución?