LABORATORIO 5

1. **Funcionamiento y sintaxis de uso de structs.**
   struct es un tipo de dato que permite combinar diferents tipos de datos en un sólo registro. sintaxis:
   struct [structure tag] {

     member definition;
     member definition;
     ...
     member definition;
   }

2. **Propósito y directivas del preprocesador.**
   Las directivas de preprocesador, como #define y #ifdef , se usan normalmente para que los programas de origen sean fáciles de cambiar y compilar en diferentes entornos de ejecución.
3. **Diferencia entre * y & en el manejo de referencias a memoria (punteros).**
   & regresa la dirección de memoria de otra variable/puntero y * guarda la dirección de una variable.
4. **Propósito y modo de uso de APT y dpkg.**
   dpkg es un gestor de paquetes pensado para debian que solo instala el packete en especifico sin tocar el packageName.deb a difrenecia de apt que instalará el paquete y cualesquiera que sean sus dependencias.
5. **¿Cuál es el propósito de los archivos sched.h modificados?**
   es preparar el ambiente para nuestro sistema de calendarización, esto definiendo estructuras y variables así como modelos de manejo de tareas entre otras cosas.
6. **¿Cuál es el propósito de la definición incluida y las definiciones existentes en el archivo?**
   Estas definen la politica de calendarización a ser utilizada.
7. **¿Qué es una task en Linux?**
   Es un proceso
8. **¿Cuál es el propósito de task_struct y cuál es su análogo en Windows?**
   Su propósito es contener la información de los procesos, su equivalente en windows vendría a ser kprocess
9. **¿Qué información contiene sched_param?**
   indica los parámetros bajo los cuales funcionará nuestro calendarizador, desde tiempos de corrida hasta prioridad de procesos.
10. **¿Para qué sirve la función rt_policy y para qué sirve la llamada unlikely en ella?**
    Sirve para definir que politica de calendarización se usará, unlikely sirve para definir los casos menos probables y por tanto de menos prioridad.

11. **¿Qué tipo de tareas calendariza la política EDF, en vista del método modificado?**
    EDF calendariza las tareas en función a su tiempo de vencimiento(deadline) teniendo prioridad aquellos procesos con las deadlines más cercanas.
12. **Describa la precedencia de prioridades para las políticas EDF, RT y CFS, de acuerdo con los cambios realizados hasta ahora.**
    a. EDF
    b. RT
    c. CFS
13. **Explique el contenido de la estructura casio_task.**
    contiene los valores de un proceso tipo SCHED_CASIO_POLICY.
14. **¿Qué indica el campo .next de esta estructura?**
    Es una forma de hacer referencia al siguiente proceso.

15. **¿Qué es y para qué sirve el tipo atomic_t? Describa brevemente los conceptos de operaciones RMW (read-modify-write) y mappeo de dispositivos en memoria (MMIO)**.
    atomic_t es un tipo de variable cuyos resultados de operaciones con estas variables siempre serán atómicos. RMW se refiere a,los operadores atómicos que permiten lectura y escritura sin condición de carrera sobre el espacio de memoria que usan. MMIO un direccionamiento de memoria que simula registros como si se tratase de la memoria principal a ojos del CPU.

16. **¿Cuándo preemptea una casio_task a la task actualmente en ejecución?**
    Siempre que no se trate de un proceso de nuestra política, dado que estos estan marcados con mayor prioridad en la política definida por nosotros.
17. **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 al algoritmo EDF para lograr aislamiento temporal.**
    El aislamiento temporal es un protocolo para el tratado de tareas que se implementó en el sistema operativo para asegurar la estabilidad tanto del sistema como del proceso, de modo que el proceso aislado posee su cantidad reservada de recursos como de espacio de memoria.

Kernel_corriendo