UNIVERSIDAD DE BUENOS AIRES - FACULTAD DE INGENIERÍA
75.08 / 95.03 - SISTEMAS OPERATIVOS - CÁTEDRA MÉNDEZ
PARCIAL

Nombre:					Nota:
Padron:		Cuatrimestre:	2/2024	Fecha: 15/11/2024	

1 - Scheduler (2 pts)

Dado el siguiente scheduler, implemente la función `elegir` para conseguir una política Round Robin (puede agregar las variables globales que crea necesarias)

```
void switch(struct p* proceso); //Asuma ya implementada

struct p {
  int status; //RUNNING, RUNNABLE, BLOCKED
}
struct p[64]; // Tabla de procesos, maximo de 64

struct p* elegir(){
 //TODO: implementar aqui
}

void scheduler() {
    for (;;) {
       struct p *candidato = elegir();
        if (candidato == NULL){
            idle();
        }else {
          candidato->status = RUNNING;
          switch(candidato);
        }
    }
}
```

Asuma que `switch` ya existe y recibe un "`struct p*`" con estado RUNNING y hace un cambio de contexto a ese proceso. Cuando el proceso se suspende, devuelve el control al scheduler (en este momento el proceso tendrá estado RUNNABLE o BLOCKED). Asuma que los procesos no terminan.

2 - Filesystem (3 pts)

- /home/sisops/a es un archivo de texto con el texto "Hola mundo"
- /home/sisops/b es un hard link al archivo a
- /home/sisops/c es un symbolic link al archivo b

a) Cuantos accesos a inodos y bloques de datos ejecutan cada una de las siguientes operaciones:
   i. cat /home/sisops/a
   ii. cat /home/sisops/b
   iii. cat /home/sisops/c

b) Mencione una ventaja de los hard links sobre los symbolic links y una ventaja de los symbolic links sobre los hard links.

3 - Locks (3 pts)

a. Cuáles son las diferencias entre un spin lock y un sleep lock. ¿Cuándo usaría cada uno?

b. ¿Es correcta la siguiente implementación de este **spinlock**? Explique qué problema tiene y cómo solucionarlo:

```
void spin_lock(int* lock) {
        while(*lock != 0){
        }
        *lock = 1;
}
```

4.1 (0.5 pts) ¿Cuál de las siguientes afirmaciones describe mejor el propósito y funcionamiento del Translation Lookaside Buffer (TLB) en un sistema de memoria virtual?

a - La TLB almacena copias de datos de la memoria principal para reducir los tiempos de acceso, similar a una caché de CPU.
b - La TLB almacena las traducciones de direcciones virtuales a físicas más frecuentemente utilizadas, reduciendo la necesidad de acceder a las tablas de páginas en la memoria principal.
c - La TLB almacena las páginas más frecuentemente utilizadas, reduciendo la necesidad de acceder a las páginas en la memoria principal.
d - La TLB almacena las tablas de páginas más frecuentemente utilizadas, reduciendo la necesidad de acceder a las tablas de páginas en la memoria principal

4.2 (0.5 pts) En un sistema operativo moderno, ¿cuál es la principal razón para mantener separado el espacio de usuario del espacio de kernel, y qué técnica utiliza el kernel para permitir que los programas en espacio de usuario accedan a recursos privilegiados sin comprometer la seguridad del sistema?

a - La separación protege al kernel de posibles errores en aplicaciones de usuario, y el acceso se permite mediante llamadas al sistema que validan las solicitudes antes de ejecutar operaciones privilegiadas.
b - La separación permite que el espacio de usuario ejecute código más rápido, y el acceso al kernel se realiza a través de interrupciones de hardware directas que garantizan la eficiencia.
c - La separación evita que las aplicaciones de usuario accedan a memoria restringida, y el kernel concede acceso mediante variables globales que almacenan información sensible.
d - La separación se utiliza principalmente para proteger la privacidad de los datos, y el acceso se permite mediante funciones públicas en el espacio de usuario que verifican permisos.

4.3 (0.5 pts) En un sistema operativo, la función `malloc` se utiliza para asignar memoria dinámica en el espacio de usuario. ¿Qué sucede internamente cuando se solicita un bloque de memoria que excede el tamaño del heap disponible actual, y qué técnica emplea el sistema para gestionar este tipo de solicitudes?

a - malloc falla y devuelve un puntero nulo, ya que no es capaz de asignar más memoria que la disponible en el heap actual.
b - malloc inicia la recolección de basura para liberar memoria y reutilizar los bloques existentes, sin necesidad de ampliar el heap.
c - malloc amplía el heap solicitando más memoria al sistema mediante la llamada sbrk o mmap, ajustando así el tamaño del espacio de memoria disponible dinámicamente.
d - malloc intercambia el contenido del heap con la memoria secundaria (disco) para liberar espacio, asignando el bloque solicitado sin afectar el tamaño del heap.

4.4 (0.5 pts) El Completely Fair Scheduler (CFS) de Linux utiliza un árbol rojo-negro ¿Qué ventaja ofrece esta estructura al objetivo del CFS de asignar CPU de forma justa?

a - Asigna CPU en orden de llegada, priorizando los procesos más antiguos en la cola de ejecución.
b - Almacena prioridades de procesos y ejecuta primero los de menor prioridad para una distribución justa.
c - Permite actualizar el proceso con menor tiempo de ejecución virtual en tiempo logarítmico, permitiendo así encontrarlo y seleccionarlo rápidamente durante un cambio de contexto.
d - Divide procesos en grupos y asigna CPU proporcional al tamaño de cada grupo.