SO2 (18/04/2024)

Primer control de laboratorio

Crea un fichero que se llame “respuestas.txt” donde escribirás las respuestas para los apartados de los ejercicios del control. Indica para cada respuesta, el **número de ejercicio y el número de apartado** (por ejemplo, 1.a, 1.b, ...).
Importante: para cada uno de los ejercicios tienes que partir del código suministrado de Zeos.

1. *(3 puntos)* Comprensión de Zeos

a) (1 punto) ¿En que dirección de memoria se encuentra la pila justo al empezar la ejecución de la rutina ´main´ de sistema? ¿Como la has encontrado? ¿Corresponde a algún proceso del vector task? ¿Cómo lo has hallado?
b) (1 punto) En el código inicial os damos el tratamiento de las excepciones ya implementado. ¿Es posible ver el código ensamblador de la rutina de tratamiento de la excepción de división por zero? Caso afirmativo muestra el código y cómo lo has hallado.
c) (1 punto) Dado un fichero *exam.*o que contiene la función *shared* implementada ¿Como debe ser **la línea (o líneas) de comandos** para linkar este objeto y que puedas usar la función tanto desde el código de sistema como de usuario?

2. *(2 puntos) Oh my fork!*

El bucle de copia de datos de usuario implica que hay que modificar entradas de la tabla de páginas (TP) del proceso actual temporalmente para mapear la zona de datos del proceso hijo. Para ahorrarnos esta modificación y dado que sólo estamos usando una entrada del directorio, queremos mapear temporalmente toda la TP del proceso hijo en la entrada 1 del directorio, y así realizar la copia de toda la zona de datos con una única llamada a *copy_data*. Implementa esta modificación del *sys_fork*.

3. *(5 puntos) Read my chars*

Queremos añadir a nuestro ZeOS una funcionalidad para leer 1 tecla del teclado :

```
int read(char* b);
```

Esta llamada bloquea al proceso actual en una lista de bloqueados en el teclado hasta que se pulse una tecla, momento en que desbloqueará al proceso y lo pondrá en ejecución, copiando la tecla leida al buffer ´b´. Si varios procesos usan esta llamada, el orden de desbloqueo tiene que seguir un orden FIFO. Esta llamada debe devolver error si el buffer no se encuentra dentro del espacio de direcciones del proceso. Esta llamada a sistema tiene que usar la interrupción 130 para realizar la entrada a sistema (en lugar de los mecanismos ya implementados) y debe ejecutar la rutina de servicio directamente pues será el único servicio accesible mediante esta interrupción. El parámetro se pasará por registro.
Implementa las funciones de sistema siguientes para gestionar la lista de bloqueados en el teclado:

void block_for_keyboard(void) : bloquea el proceso actual.

void unblock_first() : desbloquea y pasa a ejecutar el primer proceso de la lista. Si no hay procesos bloqueados esta función no hace nada.
La solución tiene que ser genérica y funcionar de forma eficiente para cualquier número de procesos.

Se pide:

a) (0,75 puntos) Implementa el código del wrapper de la llamada *read.*
b) (0,75 puntos) Implementa el código del handler de esta llamada a sistema.
c) (0,5 puntos) Indica qué estructuras de datos se tienen que añadir y/o modificar. Añade el código necesario para inicializarlas.
d) (0.5 puntos) Implementa la rutina *block_for_keyboard*.
e) (0.5 puntos) Implementa la rutina *unblock_first*.
f) (1 punto) Implementa el código de la rutina *sys_read*.
g) (1 punto) ¿Es necesario modificar alguna otra llamada a sistema o parte del sistema para implementar por completo esta funcionalidad? Si es así implementa los cambios necesarios.

4. Entrega

Sube al Racó los ficheros "respuestas.txt" junto con el código que hayas creado en cada ejercicio. Para entregar el código utiliza:

```
> tar zcfv examen.tar.gz zeos
```