CC41B Sistemas Operativos
Control 2 – Semestre Primavera 2007
Prof.: Luis Mateu

Pregunta 1

Se desea agregar locks de lectura/escritura a nSystem. Estos locks son una solución para el problema de los lectores/escritores visto en clases. La API es la siguiente:

Procedimiento	***Significado***
`void nEnterRead(nRWLock* l);`	Solicitud para leer
`void nExitRead(nRWLock* l);`	Notificación de salida de lectura
`void nEnterWrite(nRWLock* l);`	Solicitud para escribir
`void nExitWrite(nRWLock* l);`	Notificación de salida de escritura

Implemente esta API, es decir la estructura nRWLock y los procedimientos nEnterRead, nExitRead, nEnterWrite y nExitWrite, como mecanismo de sincronización nativo de nSystem. Esto significa que Ud. debe implementar esta API usando los procedimientos de bajo nivel de nSystem (START_CRITICAL, Resume, PutTask, etc.). Ud. *no puede usar* otros mecanismos de sincronización ya disponibles en nSystem, como semáforos, monitores, mensajes, etc.

Requerimiento de simplicidad: su implementación *debe* tener alguna forma de *hambruna* para lectores y/o escritores. *Indique y explique* considerando su solución la situación en la que se produce hambruna.

Pregunta 2

Parte a.- La siguiente es una implementación incorrecta del problema de los lectores/escritores usando *spin-locks*:

```
int mutex_lck= OPEN;
int write_lck= OPEN;
int readers= 0;
```

```
void enterRead() {
  if (readers==0)
    spinLock(&write_lck);
  spinLock(&mutex_lck);
  readers++;
  spinUnlock(&mutex_lck);
}
```

```
void exitRead() {
  spinLock(&mutex_lck);
  readers--;
  spinUnlock(&mutex_lck);
  if (readers==0)
    spinUnlock(&write_lck);
}
```

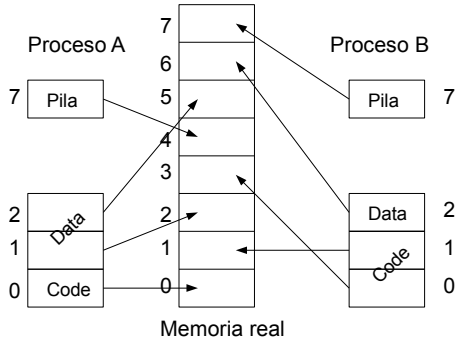
```
void enterWrite() {
  spinLock(&write_lck);
}
```

```
void exitWrite() {
  spinUnlock(&write_lck);
}
```

Haga un diagrama de threads que muestre que un lector puede entrar junto con un escritor.

Parte b.- Modifique mínimamente la solución de la parte a.- de modo que funcione correctamente (intente resolver esta parte aún cuando no resuelva la parte a.-).

Parte c.- Un sistema Unix, que usa páginas de 4 KB, ejecuta los procesos A y B. El estado actual de la memoria es el siguiente:



Haga la tabla de páginas para el proceso A indicando en la tabla: número de página virtual, número de página real y atributos de validez y escritura (V y W).

Parte d.- Suponga que el proceso A invoca sbrk pidiendo 6 KB adicionales de memoria. Modifique el diagrama de la parte c.- de modo que muestre una asignación válida de la memoria. Explique sus supuestos. (Esta pregunta tiene muchas respuestas correctas. No necesita confeccionar tablas de páginas como la que se pidió en la parte c.)

Parte e.- Vuelva a considerar como estado inicial de la memoria el diagrama de la parte c.- (es decir no considere los cambios señalados en la parte d.-). Suponga que el proceso A invoca exit (es decir el proceso termina) y luego el proceso B crea un proceso semi-ligero C. Modifique el diagrama mostrando una posible asignación de la memoria. Explique sus supuestos.