NO MODIFICAR *HAL.h* NI *const.h*

16 / 10 / 2018

INCLUSION DE UNA LLAMADA SIMPLE

* Cambios realizados en *llamsis.h*
  + Se modifica #define NSERVICIOS: 3 => 4
  + Se añade #define OBTENER\_ID
* Cambios realizados en *kernel.h*
  + Se añade la declaración de la función *int sis\_obtener\_id\_pr()*
  + La función mencionada se añade a la tabla de servicios declarada en la variable *tabla\_de\_servicios*
* Cambios realizados en *kernel.c*
  + Se implementa la función *int sis\_obtener\_id\_pr()*
* Cambios realizados en *serv.h*
  + Se declara la llamada de usuario *int obtener\_id\_pr()*
* Cambios realizados en *serv.c*
  + Se ha implementado la llamada de usuario que permite realizar esta llamada a sistema
* Cambios en *init.c*
  + Se descomenta el bloque que prueba la llamada que se ha implementado

PERDIDO PROGRESO DE LA FUNCION DORMIR()

23 / 10 / 2018

- Cambios realizados en *llamsis.h*

* Se modifica #define NSERVICIOS: 5 => 10
* Se añaden:
  + #define CREAR\_MUTEX 5
  + #define ABRIR\_MUTEX 6
  + #define LOCK\_MUTEX 7
  + #define UNLOCK\_MUTEX 8
  + #define CERRAR\_MUTEX 9

- Cambios realizados en *kernel.h*

* Se declara un struct que contiene la información de los mutex
* Se añaden las siguientes declaraciones:
  + int sis\_crear\_mutex()
  + int sis\_abrir\_mutex()
  + int sis\_lock()
  + int sis\_unlock()
  + int sis\_cerrar\_mutex()
* Se incluyen estas funciones en la tabla de servicios

- Cambios realizados en *kernel.c*

* Se implementan las funciones declaradas en *kernel.h*
* Se incluye en main la instrucción *iniciar\_tabla\_mutex()*

- Cambios realizados en *servicios.h*

* Se declaran las siguientes funciones
  + int crear\_mutex(char \*nombre, int tipo)
  + int abrir\_mutex(char \*nombre)
  + int lock(unsigned int mutex\_id)
  + int unlock(unsigned int mutex\_id)
  + int cerrar\_mutex(unsigned int mutex\_id)
* Se declaran los mismos #defines RECURSIVO, NO\_RECURSIVO para que puedan ser accedidos por los programas de usuario

- Cambios realizados en *serv.c*

* Se implementan las llamadas al sistemas del punto anterior

24 / 10 / 2018

- Cambios realizados en *kernel.h*

* Se incluye en el struct *mutex* una variables que indica si el mutex está abierto
* Se crea una nueva lista que almacena los procesos que han intentado bloquear mutex no recursivos ya bloqueados

ERRORES IMPLEMENTANDO EL MUTEX

IMPLEMENTACION DEL ROUND ROBIN

* Modificaciones en *kernel.c*

*-* Se modifican las int\_reloj() e int\_sw() para calcular el número de ciclos que se ha encontrado el proceso en ejecución. Si se llega al límite indicado por TICKS\_POR\_RODAJA, se expulsa el proceso en ejecución y se coloca al final de la lista de procesos listos

- Se modifica el planificador para que ponga a 0 el valor ticks\_en\_ejecución de cada proceso.

* Modificaciones en *kernel.h*

*-* Se añade al struct BCP el campo ticks\_en\_ejecución, que indica cuántos ciclos lleva el ciclo en ejecución desde que fue elegido por el planificador

26/10/2018

* Se han modificado *kernel.h* y *kernel.c* para obtener una mayor claridad