LAB 3
SYNCHRO IN OS161

PROGRAMMAZIONE DI SISTEMA (JZ-ZZ)

2022/23

ANDREA PORTALURI

OBIETTIVI DEL LABORATORIO

- Implementazione dei lock
- Implementazione delle condition variables

SEMAFORI IN OS161

Il semaforo è una primitiva che permette la sincronizzazione di threads/processi. Sono implementati in OS161 (debuggare *sy1* per tracciare l'esecuzione dei semafori)

```
struct semaphore *s;
s = sem_create("MySem1", 1); /* Valore iniziale uguale a 1 */
P(s) /* wait(s), entro nella sezione critica e S-- */
    critical section
V(s) /* signal(s), esco dalla sezione critica e S++ */
```

LOCK IN OS161

Il lock è una primitiva di sincronizzazione che permette la realizzazione di mutua esclusione. NON sono implementati in OS161, seppur sia disponibile l'interfaccia (*sy2* per test lock)

```
struct lock *myLock = lock_create("LockName");
lock_acquire(myLock);
    critical section
lock_release(myLock);
```

Lock introduce il concetto di ownership (il thread che acquisisce il lock, deve anche rilasciarlo)

LOCK IN OS161

Si chiede di realizzare il lock in due modi (analoghi, da gestire le opzioni per utilizzare l'uno o l'altro):

- Utilizzando i semafori (già implementati) modificando la `struct lock` in `kern/include/synch.h` e completando le funzioni `lock_create()`, `lock_destroy()`, `lock_acquire()`, `lock_release()`, `lock_do_i_hold()` in `kern/thread/synch.c`.
- Utilizzando `wchan` e `spinlock` adattandoli alla realizzazione interna dei semafori.

CONDITION VARIABLES IN OS161

Una condition variable è una primitiva di sincronizzazione che consente di attendere che una condizione (eventualmente falsa al momento attuale) diventi vera. È sempre accompagnata da un lock, passato come parametro alle funzioni, che ne garantisce l'accesso protetto in mutua esclusione.

- `cv_wait()`: rilascia il lock (che deve essere acquisito in precedenza dal thread chiamante) e si mette in attesa di `cv_signal()` o `cv_broadcast()`.
- `cv_signal` e `cv_broadcast()`: svolgono lo stesso compito ma differiscono nel numero di thread svegliati (uno e tutti, rispettivamente).

CONDITION VARIABLES IN OS161

Si richiede la realizzazione della condition variable mediante wait_channel e spinlock. ATTENZIONE: `wait_channel` ha una semantica simile a quella della condition variable ma ha uno spinlock invece del lock.

I programmi di test che utilizzano le condition variables sono sy3 e sy4, che chiamano le funzioni `cvtest()` e `cvtest2()`.

Le definizioni delle funzioni sulle condition variables sono in `kern/include/synch.h` e `kern/thread/synch.c`.