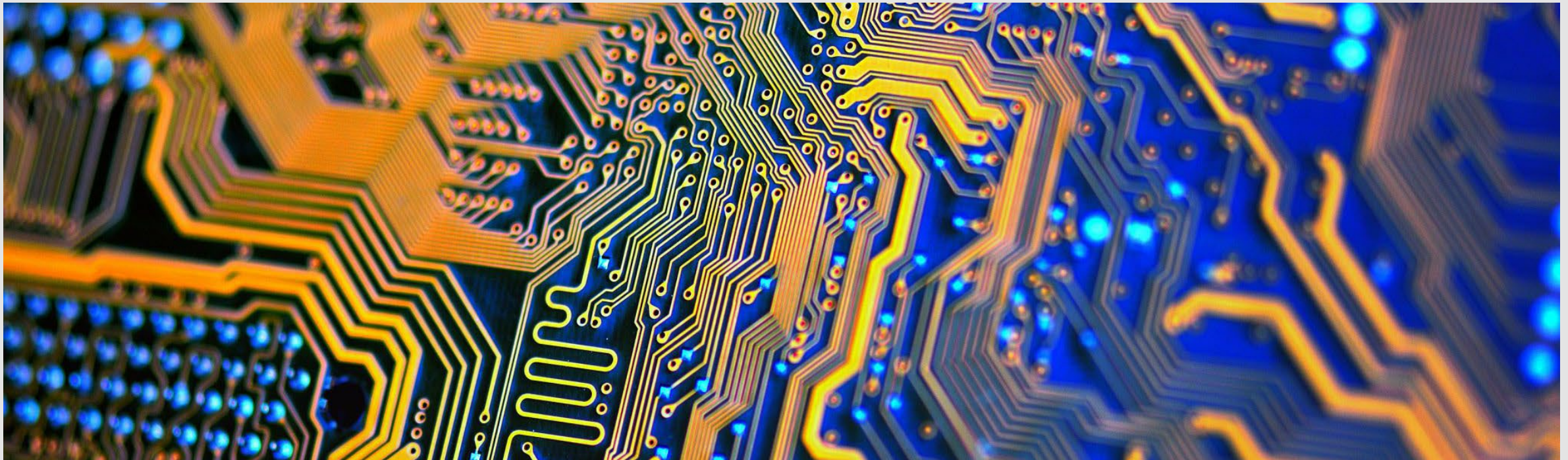


PROGRAMMAZIONE DI SISTEMA
A.A. 2022/2023

S318904 MARCELLO VITAGGIO
S317264 FABRIZIO VITALE

IMPLEMENTAZIONE SCHEDULER AVANZATO – PRIORITY SCHEDULING



OPERAZIONE SEMAPHORE DOWN IN SYNCH.C

- File: threads/synch.c
- Modifiche: I thread vengono aggiunti alla lista di attesa del semaforo utilizzando l'inserimento ordinato in base alla priorità.

```
- list_push_back (&sema->waiters, &thread_current ()->elem);  
+ list_insert_ordered(&sema->waiters, &thread_current ()->elem,  
thread_higher_priority, NULL);
```

OPERAZIONE SEMAPHORE UP IN SYNCH.C

- File: threads/synch.c
- Modifiche: I thread vengono aggiunti alla lista di attesa del semaforo utilizzando l'inserimento ordinato in base alla priorità.

```
+ struct thread *t = NULL;
+ if (!list_empty (&sema->waiters)){
+   list_sort(&sema->waiters, thread_higher_priority, NULL);
+   t = list_entry (list_pop_front (&sema->waiters), struct thread, elem);
+   thread_unblock (t);
+ }
+ if (t != NULL && t->priority > thread_current()->priority) {
+   thread_yield_to_higher_priority();
+ }
```

MODIFICHE ALL'OPERAZIONE LOCK IN SYNCH.C

- File: threads/synch.c
- Modifiche:
 - Implementata la donazione di priorità in lock_acquire.
 - La priorità donata viene rivalutata e la lista dei pronti viene ordinata in lock_release.

```
+ if (lock->holder != NULL && donor->priority > lock->holder->priority) {  
+   /* Priority donation logic */  
+ }  
+  
+ /* Priority re-evaluation and sorting the ready list during release */  
+ if(!list_empty(&lock->semaphore.waiters)){  
+   /* Release donation logic */  
+ }
```

CREAZIONE THREAD CON CESSIONE DI PRIORITÀ IN THREAD.C

- File: threads/thread.c
- Modifiche: Ora i thread cedono immediatamente la priorità a thread con priorità superiore appena creati, se necessario.

```
+ if (priority > thread_current()->priority) {  
+   thread_yield_to_higher_priority();  
+ }
```

DONAZIONE DI PRIORITÀ NEI THREAD IN THREAD.C

- File: threads/thread.c
- Modifiche:
 - `recompute_thread_priority` viene utilizzata per calcolare la priorità effettiva considerando le donazioni.
 - `sort_ready_list` viene utilizzata per mantenere la lista dei pronti ordinata per priorità.

```
+void recompute_thread_priority (struct thread*);  
+void sort_ready_list();
```

IMPOSTAZIONE PRIORITÀ THREAD IN THREAD.C

- File: threads/thread.c
- Modifiche: `thread_set_priority` ora regola la priorità del thread considerando la donazione di priorità e può cedere il controllo se necessario.

```
+ if (cur->priority != cur->base_priority) {  
+   cur->base_priority = new_priority;  
+ }  
+ else {  
+   cur->priority = new_priority;  
+   cur->base_priority = new_priority;  
+ }  
+ sort_ready_list();  
+ thread_yield_to_higher_priority();
```

ESTENSIONE DELLA DONAZIONE DI PRIORITÀ A THREAD.H

- File: threads/thread.h
- Modifiche: Aggiunge strutture e funzioni relative alla logica di donazione di priorità.

```
+struct semaphore sema;  
+int base_priority;  
+struct thread *donee;  
+struct list donorList;  
+struct lock *wantsLock;  
+struct list_elem donationElem;
```