Tecnologie software

- Esercitazione finale
- Libreria thread in user space: si scriva una libreria che consenta di creare e schedulare thread in user space (senza usare le chiamate di sistema). La libreria non gestisce la pre-emption, per cui ogni thread rilascia la CPU volontariamente perche' termina oppure chiamando una opportuna funzione "yield".

API proposta

- THREAD
  - Create(&id, function_pointer)
  - Terminate(id)
  - Exit()
  - Yield()
- SEMAFORI
  - Create(&id)
  - Up(id)
  - Down(id)
  - Destroy(id)
- SCHEDULING
  - Scheduler()

Tabella dei thread (esempio)

- const int MAX_THREAD= 10;

```
struct thread_item {
int thread_id;
jmp_buf env;
int  thread_state;
}
```

setjmp/longjmp (pseudocodice)

```
int a=4;
Int b =6;
B+=a;
jmp_buf env;
-> int ret = setjmp(&env);
if (ret == 0) {
}
else
if(ret == 5) {
…
}


….
….

longjmp(env, 5);
```

Thread state

- Suspend
- Running
- Blocked
- Ready

- Roundrobin

Test

- Produttore-consumatore

Main()
Init_library() (setjmp())
-Create()
-Create()
…
…
…
Yield()

-Terminate()
…

Consegna

- 30 Giugno 2010

- Dopo questa data:
  - Punteggio ridotto
  - Aiuto minimo