

/* link testo <https://www.cs.unibo.it/~renzo/so/compiti/2022.06.21.tot.pdf> */

Esercizio c1

```
monitor collocaimento {
    typedef struct {
        char* nome;
    } persone;

    persone waiting_p[skill]; // lista FIFO di persone in attesa di essere assunti con sk
    int waiting_dat[skill]; // numero di datori di lavoro in attesa di qualcuno con ski
    char *name2send = null;
    condition p, dat;

    void cercolavoro(char *nome, char *skill){
        // controllo se ci sono richieste per la skill
        if (waiting_dat[skill]==0){
            p.wait();
        }
        // aggiungo la nuova persona alla lista di attesa
        while (!waiting_p[skill].isEmpty())
            waiting_p[skill].next();
        waiting_p[skill].nome = nome;
        name2send = waiting_dat[skill].nome;
        waiting_dat[skill]--;
        dat.signal();
    }

    char* assumo(char *skill){
        if (waiting_p[skill].isEmpty())
            dat.wait();
        // qualcuno è in attesa con la skill richiesta
        if (name2send == null){
            name2send = waiting_p[skill].nome;
            waiting_p[skill].remove(); // rimuovo il primo elemento in cima che ho salvato p
        }
        p.signal(); // libero p
        return name2send;
    }
}
```

Esercizio c2

```
server[N] = ...pids
serversender[N] = [None, ..., None]
```

```
process dispatcher:
    while true:
        <msg, mitt> = arecv(ANY)
        # da completare
```

Esercizio g1

- monoprocesso
- scheduler multilivello a 2 livelli
 - processi alto priorità -> algo FIFO
 - processi bassa priorità -> round-robin con quantum = 3ms ALTA PRIORITÀ:
- processi periodici: h e k riattivati ogni 6ms
 - 1ms elaborazione -poi-> terminano BASSA PRIORITÀ:
- P: 4ms elaborazione -> 2ms i/o -> 2ms elaborazione -> 1ms i/o -> 5ms elaborazione
- Q: 5ms elaborazione -> 3ms i/o -> 2ms elaborazione -> 1ms i/o -> 4ms elaborazione

tempo| 0 1 2 5 7 8 9 10 13141516 1819 212223 26 282930 32 CPU | h|k|-p|q-
|h|k|p|-q-|p|h|k|q-|p|q-|h|k|-p-|q-|h|k|p-| I/O | |p-| |-q-| |q|p| Per calcolare questo
diagramma abbiamo innanzitutto distribuito i processi in base FIFO di alta
priorità: 1ms di esecuzione ogni 6ms di attesa