Liste di processi

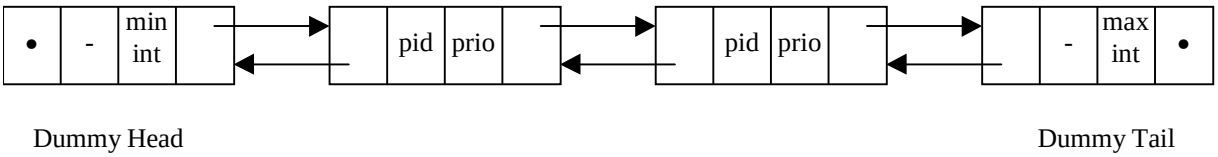
Ogni processo può trovarsi in al più una tra:

- ready list
- sleep list
- liste associate ai semafori

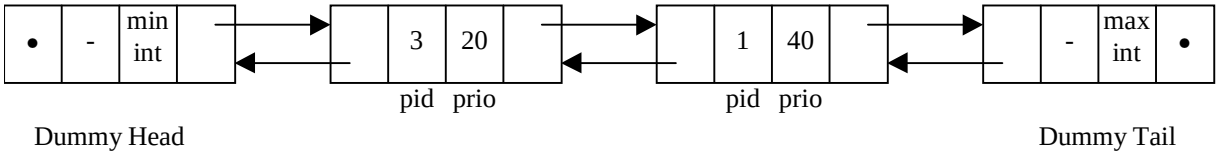
Criteri di progetto:

- liste concatenate (*linked*): inserimento/cancellazione veloci
- codice unificato → interfaccia e implementazione generali:
  - liste con priorità (anche se non necessaria per tutte)
  - liste doubly linked (accesso a elemento precedente veloce)
  - accesso veloce all'ultimo elemento (oltre che al primo)
  - testa e coda = elementi dummy (evita *if* (*testa*==NULL) …)

Concettualmente: liste di elementi (process id, priorità):



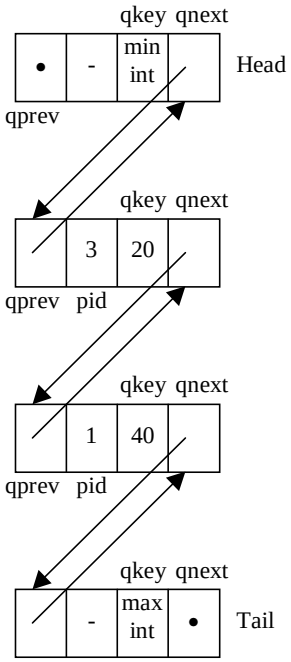
Implementazione



- elementi liste sono dati statici (elementi array q) e non dinamici:
  - minori dipendenze (gestione liste ↔ gestione memoria)
  - implementabile con compilatore/linguaggio con modello di memoria semplice (senza heap)
  - maggior velocità (risparmia chiamate `malloc()` etc.)
- puntatori logici → interi (indici array)
- *pid* implicito, non memorizzato esplicitamente in elementi lista:
  - *pid k* (`0`≤*k*≤`NPROC-1`) memorizzato in elemento array `q[`*k*`]`
  - possibile perché ogni *pid* sta al più in una lista
  - accesso a dato *pid* è O(1) (sarebbe O(|lista concatenata|)
- head/tail dummy in elementi di array q oltre `0..NPROC-1`

Array q

	qkey	qnext	qprev
0			
1	40	33	1
2			
3	20	1	32
4			
	. . .	. . .	. . .
NPROC-1			
NPROC-1+1			
	. . .	. . .	. . .
(Head) 32	MININT	3	NULL
(Tail) 33	MAXINT	NULL	1
	. . .	. . .	. . .
NPROC-1+2(N.Max Liste)			



Header file q.h

```
/* q.h - firstid, firstkey, isempty, lastkey, nonempty */

/* q structure declarations, constants, and inline procedures */

#ifndef NQENT
#define NQENT NPROC + NSEM + NSEM + 4 /* for ready & sleep */
#endif

struct  qent {    /* one per process plus two per list    */
   int  qkey;     /* key on which the queue is ordered    */
   int  qnext;    /* pointer to next process or tail      */
   int  qprev;    /* pointer to previous process or head  */
   };

extern  struct qent q[];    /* array q */
extern  int nextqueue;

/* inline list manipulation procedures */

#define isempty(list)   (q[(list)].qnext >= NPROC)
#define nonempty(list)  (q[(list)].qnext < NPROC)
#define firstkey(list)  (q[q[(list)].qnext].qkey)
#define lastkey(tail)   (q[q[(tail)].qprev].qkey)
#define firstid(list)   (q[(list)].qnext)

#define EMPTY       (-1)   /* NULL pointer */

extern char *deq(int);
extern char *seeq(int);
```