

Fondamenti di Informatica e Laboratorio T-AB

Appello del 6 Luglio 2009

Progetto e programmazione in C

Paolo Torroni, Rebecca Montanari, Marco Montali

Università degli Studi di Bologna, Facoltà di Ingegneria
Lauree nei settori di Ingegneria dell'Informazione

Anno Accademico 2008/2009

Progetto e programmazione in C (pagina 1/3)

Scenario:

- ▶ *In molte applicazioni è necessario gestire delle **code**: di clienti da servire, di processi in attesa di CPU, di pratiche da sbrigare, di progetti da pianificare, etc.*
- ▶ *La più comune politica di gestione delle code è detta **FIFO** (first-in-first-out). Secondo tale politica, viene servita ogni volta la richiesta che è in attesa di servizio da più tempo.*
- ▶ *Con una politica di questo tipo, spesso si rendono necessari dei meccanismi per consentire alle richieste più urgenti di “saltare” la coda: per questo vengono assegnate delle **priorità** alle richieste.*

NOTE:

♡ Tempo a disposizione per questa parte: **2 ore e 30 minuti**.

♣ Il progetto della soluzione può essere presentato come commento in cima al file che contiene il `main()`. Curarsi di documentare il codice tramite commenti.

◇ Consegnare tutti i file sorgenti (.c, .h) ed eventuali altri file tramite *esamix*.

♠ **È consentita la consultazione** di materiale di riferimento. **È consentito il riuso** di codice proprio o reperito in rete. **Non è consentito** l'uso dell'e-mail né di altri mezzi di comunicazione, durante lo svolgimento della prova d'esame.

Progetto e programmazione in C (pagina 2/3)

Domanda: Si progetti un tipo di dato astratto **coda** i cui elementi sono dei record (tipo **elem**) contenenti:

- ▶ una stringa di testo (**identifier**);
- ▶ un valore intero (**priority**);
- ▶ un intero lungo (**timestamp**).

La coda deve essere gestita secondo una politica FIFO con priorità:

- ⊙ L'**identifier** è una stringa qualsiasi che identifica l'elemento (es: "primo elemento", "secondo elemento", ...).
- ⊙ Il valore 0 di **priority** indica massima priorità, il valore 1 priorità minore, il 2 ancora minore, etc. fino a 255 (minima priorità).
- ⊙ Il **timestamp** è il momento in cui un elemento viene inserito in coda, misurato in secondi a partire da un tempo iniziale 0.

I **timestamp** dei vari elementi possono essere non consecutivi tra loro, ma sono ordinati nel momento in cui vengono registrati.

Progetto e programmazione in C (pagina 3/3)

In particolare, le istanze di tipo **coda** sono gestite mediante le seguenti funzioni:

- ▶ **new_q** (*costruttore*: crea una coda vuota);
- ▶ **free_q** (*distruttore*: libera la memoria dalla coda);
- ▶ **view_q** (visualizza tutti gli elementi in coda);
- ▶ **put_elem** (aggiunge un elemento in coda, leggendo da tastiera i valori **identifier**, **priority** e **timestamp**);
- ▶ **get_elem** (restituisce l'elemento con **priority** più vicina a 0, e a parità di priorità, quello con **timestamp** più basso);
- ▶ **remove_elem** (elimina dalla coda l'elemento con **priority** più vicina a 0, e a parità di priorità, con **timestamp** più basso);
- ▶ **save_q** (salva la coda su file);
- ▶ **read_q** (legge la coda da file).

Si progetti l'interfaccia e si dia un'implementazione del tipo **coda**, facendo le ipotesi che si ritengano più opportune.