

**Sistemi Operativi**  
prova di laboratorio  
– 7 ottobre 2022 –

Creare un programma **yet-another-auction-house.c** in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

**yet-another-auction-house <auction-file> <num-bidders>**

Il programma dovrà gestire un certo numero di aste descritte nel file di testo indicato e creando *num-bidders* processi figli che fungeranno da offerenti per ogni asta con offerte casuali. Il file di testo, per ogni asta da organizzare, avrà una riga del tipo *<object-description>, <minimum-offer>, <maximum-offer>*; vedere anche il file di esempio più avanti. Ogni offerente, una volta ricevute tali informazioni, formulerà una propria offerta numerica intera casuale compresa tra 0 (zero) e *maximum-offer*.

Il programma una volta avviato si istanzierà in un processo che chiameremo **J** (giudice): questo al suo avvio creerà le strutture di IPC necessarie e *num-bidders* processi figli: **B1**, **B2**, ... , **Bn**. Tutti i processi comunicheranno usando un coda di messaggi.

Il processo **J**, per ogni descrizione di asta letta dal file, dovrà lanciare la stessa inviando un messaggio a tutti gli offerenti contenente informazioni come *<auction-number>*, *<object-description>*, *<maximum-offer>*. Questi procederanno, dopo una pausa random tra 0 e 3 secondi, a fare la loro offerta attraverso un messaggio destinato **J**. Ricevute tutte le offerte il processo **J** determinerà l'esito dell'asta come: "conclusa con successo" o "andata a vuoto" se nessuna delle offerte ha superato il valore minimo. Nel primo caso dovrà anche individuare l'offerente migliore tenendo anche conto, in caso di ex aequo, dell'ordine d'arrivo dell'offerta (il primo a farla ha precedenza). Ogni processo dovrà riportare a schermo opportune informazioni sui passi compiuti e le informazioni scambiate: vedi esempio a seguire.

Concluse tutte le aste, tutti i processi dovranno terminare correttamente e spontaneamente e, sempre il processo **J**, dovrà occuparsi di ripulire le strutture persistenti di IPC.

Suggerimento: usare *rand()* per campionare un numero pseudo-casuale e, una tantum, *srand(time(NULL))* da *time.h* per ottenere sequenze di numeri di volta in volta diversi.

**Tempo:** 2 ore

Un file di aste tipo potrebbe essere il seguente:

Cotta di Mithril,100,500  
Palantír,700,1000  
Anelli degli Uomini,50,600  
Fiala di Galadriel,90,200  
Unico Anello,9000,10000  
Anelli dei Nani,100,600  
Anelli degli Elfi,400,700

L'output tipo di una esecuzione potrebbe essere il seguente:

\$ ./yet-another-auction-house auctions.txt 4

J: lancio asta n.1 per Cotta di Mithril con offerta minima di 100 EUR e massima di 500 EUR

B2: aspetto per 0 secondi

B2: invio offerta di 334 EUR per asta n.1 per Cotta di Mithril

J: ricevuta offerta da B2

B1: aspetto per 1 secondi

B1: invio offerta di 221 EUR per asta n.1 per Cotta di Mithril

J: ricevuta offerta da B1

B4: aspetto per 1 secondi

B4: invio offerta di 402 EUR per asta n.1 per Cotta di Mithril

J: ricevuta offerta da B4

B4: aspetto per 3 secondi

B3: invio offerta di 23 EUR per asta n.1 per Cotta di Mithril

J: ricevuta offerta da B3

J: l'asta n.1 per Cotta di Mithril si è conclusa con 3 offerte valide su 4; il vincitore è B4 che si aggiudica l'oggetto per 402 EUR

J: lancio asta n.2 per Palantír con offerta minima di 700 EUR e massima di 1000 EUR

B4: aspetto per 3 secondi

B2: invio offerta di 409 EUR per asta n.2 per Palantír

J: ricevuta offerta da B2

B4: aspetto per 0 secondi

B1: invio offerta di 99 EUR per asta n.2 per Palantír

J: ricevuta offerta da B1

B4: aspetto per 3 secondi

B4: invio offerta di 598 EUR per asta n.2 per Palantír

J: ricevuta offerta da B4

B3: aspetto per 2 secondi

B3: invio offerta di 650 EUR per asta n.2 per Palantír

J: ricevuta offerta da B3

J: l'asta n.2 per Palantír si è conclusa senza alcuna offerta valida pertanto l'oggetto non risulta assegnato

[...]

J: sono state svolte 7 aste di cui 5 andate assegnate e 2 andate a vuoto; il totale raccolto è di 3344 EUR