

Principi di Sistemi Operativi – Esercitazione 4

1- Traghetto (Esame del 31/03/2008)

Due isole (A e B) sono collegate tra loro da un servizio di traghetto per trasporto auto. Un unico traghetto viaggia continuamente da un'isola all'altra, partendo da un molo e arrivando all'altro molo.

Le auto si imbarcano ad uno dei due moli (A o B) e vengono trasportate fino al molo opposto. Ogni auto ha un proprio peso ($0 < p \leq P$) che deve essere indicato al momento dell'imbarco nel molo di partenza. Il traghetto contiene al più MAX auto e non può comunque sopportare un carico totale maggiore di K (con $K > P$). Le auto aspettano la disponibilità del traghetto presso il loro molo e vengono caricate fino alla massima capienza del traghetto. Una volta accettate a bordo, queste aspettano finché la traversata si conclude.

All'arrivo a destinazione, le auto vengono scaricate e girano all'interno dell'isola (non è richiesto che tornino indietro all'isola di partenza!).

Il traghetto parte inizialmente da uno dei due moli (si scelga a piacere quale), carica le auto che aspettano a quel molo e parte per il viaggio. Arrivato a destinazione, scarica le auto a bordo e carica quelle in attesa al molo. Il traghetto non riparte se non ci sono almeno MIN auto salite a bordo.

Si implementi una soluzione usando il costrutto monitor per modellare il **servizio di traghetto** tra le due isole e i processi per modellare le **auto** e il **traghetto** e si descriva la sincronizzazione tra i processi. Nella soluzione si massimizzi l'utilizzo delle risorse. Si discuta se la soluzione proposta può presentare starvation e in caso positivo per quali processi, e si propongano modifiche e/o aggiunte per evitare la starvation.

2- Elezione del sindaco (Esame 25/06/2007)

In una città vivono **E cittadini** che eleggono periodicamente il loro sindaco in una rosa di **C candidati** (con C costante per ogni tornata elettorale). Ogni cittadino si reca al seggio elettorale (unico in città) e aspetta di poter esprimere la sua preferenza di voto (scegliendo il codice del candidato *da 0 a C-1*). Essendoci un solo seggio a disposizione, solo un elettore per volta può entrarvi per votare. Si faccia, altresì, in modo che il voto di uno stesso cittadino possa variare da elezione ad elezione. Il processo di elezione del sindaco si svolge sotto la supervisione di un **prefetto**, il quale si occupa di raccogliere i voti e procedere alla selezione di un nuovo sindaco tra i C candidati. Il prefetto inizia la raccolta dei voti, solamente se non ci sono elettori che stanno utilizzando il seggio per votare e se si è raggiunto il *quorum (metà + 1 degli aventi diritto)*. Si faccia attenzione ad impedire che i cittadini possano continuare a votare dopo l'inizio della raccolta da parte del prefetto. Dopo aver conteggiato i voti assegnati a ciascun candidato, il prefetto nomina il nuovo sindaco e lo abilita a governare per **G** giorni. Il sindaco vincente è colui che conquista il maggior numero di voti nella tornata elettorale (per semplicità, a parità di punteggio, si scelga il primo candidato nell'ordine). Durante la durata in carica del sindaco, gli elettori non possono entrare nel seggio per votare, ma devono aspettare che il sindaco in carica termini il suo mandato.

Si implementi una soluzione usando il costrutto monitor per modellare il **seggio di elezione del sindaco** e i processi per modellare gli **elettori**, i **candidati a sindaco** ed il **prefetto**.