

Nome: Cognome: Compito: **18** Turno: **1**

PRINCIPI DI SISTEMI OPERATIVI

(A.A. 11-12)

15 Febbraio 2012

IMPORTANTE:

1. Si considerano parte integrante delle soluzioni i **COMMENTI *significativi*** introdotti per facilitare la lettura del codice: come tali, essi influenzano la votazione finale. Tuttavia, i messaggi di debug (ad es. le `println()`) del programma **NON SONO CONSIDERATI E QUINDI NON INFLUENZANO LA VOTAZIONE FINALE.**
2. Il tempo a disposizione è di 90 minuti.
3. Il compito deve essere svolto **solamente** nel linguaggio Java, usando le classi del package **monitor** e lavorando con l'ambiente di sviluppo **IBM Eclipse**.
4. Seguire le seguenti regole per lo svolgimento dell'esame al laboratorio base:
 - Fare il login in Linux con il proprio account (numero di tesserino e password di posta elettronica)
 - Aprire un terminale
 - Eseguire i seguenti comandi shell:
 - a. `mkdir rhome` per creare una directory locale 'di appoggio'
 - b. `sshfs USER@155.185.31.1XX:/home/nUSER rhome` (dove USER è il numero di tesserino universitario e XX è il numero della macchina fisica LINFA) per montare la propria home directory remota che risiede sul server virtuale (`lica.lab.unimo.it`) sulla directory locale 'di appoggio'Questo comando richiederà per essere eseguito di inserire la password di posta elettronica
 - Aprire Elipse (comando "eclipse" sempre da shell) e scegliere come workspace nella propria home directory locale la cartella rhome sulla quale avete appena montato la directory remota (`workspace = user/rhome`)
 - Creare un progetto Java con nome "ESAME150212-1-18" e scrivere le classi Java della soluzione nel package di default (senza nome) di tale progetto. Fare attenzione a scrivere correttamente il nome del progetto, con maiuscole e minuscole a posto!
 - Installare le classi del monitor Java e gli eventuali template
 - Finito il vostro esame (o allo scadere del tempo), dovete salvare tutto (si consiglia di salvare spesso per non perdere il proprio lavoro), chiudere Eclipse, fare il logout, lasciare il vostro PC e procedere alla consegna del testo.

In un'Ambasciata lavorano N **addetti** ed un **ambasciatore**. In ambasciata si recano le persone che abitano fuori dal loro paese d'origine e devono richiedere un visto.

Una volta arrivate in ambasciata, le persone devono recarsi da uno degli addetti (il primo libero) per verificare la correttezza dei documenti portati (tempo random deciso dall'addetto). Finita la verifica con l'addetto, devono recarsi dall'ambasciatore per richiedere la vidimazione del visto.

L'ambasciatore, una volta ricevuta la persona, controlla nuovamente la richiesta (tempo random deciso dall'ambasciatore) e, se è corretta, firma il visto e la persona può tornare a casa. Se invece l'ambasciatore trova anomalie nella richiesta* rimanda la persona da un addetto per ottenere i timbri mancanti prima di poter tornare da lui e richiedere nuovamente la firma del visto. Per l'accesso agli addetti, le persone che devono ottenere i timbri dagli addetti hanno la priorità rispetto a quelle che vengono dall'esterno.

Si implementi una soluzione usando il costrutto monitor per modellare l'**Ambasciata**, e i processi per modellare gli **addetti**, l'**ambasciatore** e le **persone**. 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.

* Si modelli la scelta da parte dell'ambasciatore se firmare o no il visto, come una scelta random da parte dell'ambasciatore stesso.