

Nome: Cognome: Compito: Turno:

PRINCIPI DI SISTEMI OPERATIVI

(A.A. 10-11)

14 Settembre 2011

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@lica.lab.unimo.it:/home/nUSER rhome` (dove USER è il numero di tesserico universitario) 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 Eclipse (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 "ESAME130711-TURNO-NUMERO" 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 **Call Centre** vi sono **M risponditori automatici** e lavorano **N tecnici** (con $N < M$). **T tecnici** (con $T < N$) lavorano anche per il reparto amministrativo.

Al Call Centre arrivano le chiamate dei **clienti** che vogliono risolvere i loro problemi. I clienti, se trovano un risponditore automatico libero, cercano inizialmente di risolvere il problema tramite questo. Dopo un tempo random (deciso dal risponditore automatico), il risponditore termina la sua parte e il cliente decide (in maniera random) se la risposta è stata soddisfacente. Se la risposta al problema è stata soddisfacente, il cliente termina la chiamata, altrimenti la chiamata viene inoltrata ad un tecnico. Se il cliente ha ancora un problema di natura amministrativa, la chiamata deve essere inoltrata ad un tecnico che lavori anche per il reparto amministrativo.

I tecnici che lavorano anche al reparto amministrativo danno la precedenza ai clienti che hanno dei problemi amministrativi, rispetto ai clienti che hanno problemi tecnici.

Dopo un tempo random (deciso dal tecnico), la chiamata si può considerare conclusa e il problema risolto.

Si implementi una soluzione usando il costrutto monitor per modellare il **Call Centre** e i processi per modellare i **tecnici**, i **risponditori automatici** e i **clienti**. 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 proponano modifiche e/o aggiunte per evitare la starvation.