Specifiche progetto Sistemi Operativi sessione autunnale A.A. 2019/2020
Docente Prof. Emanuele Lattanzi

1. Specifiche generali

Scrivere un programma multithread che consenta di valutare le performance, in termini di numero di page fault, di algoritmi di rimpiazzamento delle pagine per la gestione della memoria virtuale.
Il programma dovrà essere costituito da un oggetto principale che operi come una Memory Mangement Unit (MMU) e da un numero arbitrario (*n*) di thread ove ogni thread emuli un singolo processo (PROCESSO).
Il programma dovrà simulare una sessione di lavoro nella quale sono presenti *n* processi che possono accedere alla memoria. I processi dovranno generare indirizzi di memoria casuali e il programma terminerà una volta raggiunto il numero prestabilito di accessi in memoria totali (*accessiTotali*).

2. Specifiche dell'oggetto MMU

- L'oggetto MMU sarà la base del sistema ed utilizzerà un algoritmo di rimpiazzamento delle pagine scelto dallo studente tra FIFO, LRU, LFU, MFU e Algoritmo di seconda chance.
- Essa dovrà essere istanziata dal thread principale (main) e sarà gestita come oggetto condiviso tra i diversi thread dei PROCESSI.
- In particolare essa definirà le seguenti caratteristiche:
  - Spazio di indirizzamento virtuale 4096 byte
  - Dimensione pagina 4 byte
- La dimensione della memoria fisica sarà un parametro di lancio (*memFisica*) e, chiaramente, dovrà essere multipla della dimensione della pagina.
- La MMU dovrà, inoltre, contenere la lista delle tabelle della pagine ciascuna relativa ad un PROCESSO. La definizione della specifica delle tabelle delle pagine sarà lasciata allo studente.
- I PROCESSI dovranno generare indirizzi verso la MMU invocandone un metodo pubblico.
- L'accesso a tale metodo dovrà garantire la muta esclusione (un solo PROCESSO alla volta potrà invocare il suddetto metodo.
- La muta esclusione dovrà essere ottenuta utilizzando le primitive di sincronizzazione messe a disposizione dal linguaggio di programmazione eccezion fatta per il costrutto monitor (per chiarezza non sarà possibile utilizzare il costrutto *synchronized*) .
- Il metodo in questione processerà la richiesta andando ad interrogare l'opportuna tabella delle pagine. Nel caso in cui, l'indirizzo generato sia relativo ad una pagina già mappata in memoria la MMU si limiterà ad incrementare un contatore statistico di *page-hit*. D'altra parte, nel caso in cui l'indirizzo generato dal PROCESSO faccia riferimento ad un indirizzo contenuto in una pagina non mappata in memoria la MMU dovrà eseguire i seguenti passi:
  1) incrementare un contatore di *page-faults*
  2) se non c'è più spazio in memoria: selezionare una pagina mappata da rimuovere (vittima) secondo la politica dell'algoritmo oggetto di test (il rimpiazzamento sarà sempre di tipo GLOBALE indipendentemente dall'algoritmo di rimpiazzamento selezionato.
  3) aggiornare le opportune tabelle delle pagine
     a. indicare come non valida la pagina che è appena stata rimossa
     b. inserire una nuova entry nella tabella delle pagine del PROCESSO che ha generato il page fault (la pagina è stata appena caricata)

3. Specifiche dell'oggetto Processo

- L'entità Processo dovrà emulare un processo in esecuzione su di un sistema multiprogrammato. Ogni processo dovrà essere emulato attraverso l'utilizzo di un singolo thread.
- Ogni processo dovrà inizializzare il generatore di numeri casuali con il proprio indice numerico assegnato in modo progressivo all'atto della creazione
- Il comportamento di ogni entità Processo può essere descritto dai seguenti passi:
    - Finché non è stato raggiunto il numero complessivo di accessi in memoria
        - Effettua un richiesta per l'accesso in memoria il cui indirizzo dovrà essere scelto casualmente tra gli indirizzi logici possibili [0, 4095].
        - Quando la richiesta per l'accesso in memoria ritorna (richiesta evasa) il processo andrà in sleep per un tempo casuale nell'intervallo [5, 15] ms.

4. Specifiche di sincronizzazione ed implementazione:

- Il progetto deve essere sviluppato in ambiente Linux/Windows/macOS utilizzando **esclusivamente il linguaggio java su piattaforma Netbeans.**
- Il progetto deve essere esente da deadlock.
- La muta esclusione, l'accesso a variabili e a metodi condivisi e l'attesa dei processi dovranno essere ottenute utilizzando le primitive di sincronizzazione messe a disposizione dal linguaggio java (ad esempio: package java.util.concurrent.) ad esclusione del costrutto monitor (per chiarezza non sarà possibile utilizzare il costrutto *synchronized*).

5. Specifiche dati in input:

- Il programma dovrà ricevere in input, tramite riga di comando o tramite file di specifica, i seguenti parametri di input:
- Numero di processi (*n*)
- Numero totale di accessi in memoria (*accessiTotali)*
- Dimensione della memoria fisica (*memFisica*)

6. Specifiche dati di output:

- Al termine della simulazione il programma dovrà scrivere, su di un file di testo oppure sulla console, il numero totale dei page hit, page fault, e di rimpiazzamenti effettuati, e, per ogni processo, il numero di pagine accedute, il numero di page hit, il numero di page fault ed il numero di rimpiazzamenti effettuati.

7. Documentazione

- Il progetto Netbeans deve essere corredato dalla seguente documentazione:
    - Eventuali file di configurazione dell'applicativo
    - Relazione in formato PDF contenente:

- Identificazione studente/i: Nome cognome e matricola.
- Descrizione della progettazione: si descrivano i motivi che hanno portato alle attuali scelte progettuali. Ci si può avvalere anche di schemi a blocchi o figure.
- Descrizione dell'implementazione: si descrivano le scelte implementative anche di natura tecnica.
- Contributo degli studenti (solo nel caso di progetto svolto in coppia): per ogni studente si dovrà descrivere il ruolo avuto nel gruppo ed il relativo contributo specificando, ad esempio, quale parte del progetto/documentazione è stata implementata/redatta.
- Testing e commenti ai risultati ottenuti: la relazione dovrà contenere una sezione dedicata al test del programma dove si dovranno commentare e riportare, graficandoli, l'andamento del numero totale di page hit, page fault, e rimpiazzamenti di pagina al variare del numero dei processi (n), e al variare della dimensione della memoria fisica.
- La relazione **NON deve contenere l'intero listato del codice**.
- La relazione **NON deve superare le 10 pagine** a pena bocciatura del progetto.

8. Modalità di presentazione del progetto:

- Il presente progetto può essere presentato individualmente o in coppia, fino ad **UNA SETTIMMANA prima dell'appello** d'esame in cui s'intende sostenere la prova orale. Pertanto le scadenze di presentazione sono le seguenti:
  - **01-09-2020 ore 12.30** per chi vuole sostenere la prova orale al primo appello
  - **16-09-2020 ore 12.30** per chi vuole sostenere la prova orale al secondo appello.
- La consegna del progetto dovrà avvenire esclusivamente attraverso la piattaforma blended.uniurb.it accendendo alla sezione "Consegna progetto d'esame sessione autunnale 2019-2020". Gli studenti che non avessero accesso all'anno accademico 2019-2020 sulla piattaforma blended.uniurb.it potranno richiedere l'attivazione alla segreteria inviando una mail a: anya.pellegrin@uniurb.it
- La consegna dei progetti fatti in coppia avverrà da parte di uno solo degli studenti.
- Il progetto deve essere caricato sulla piattaforma blended.uniurb.it sotto forma di file compresso in modalità tar.gz o zip contenente il progetto Netbeans e la relazione in formato PDF.
- Progetti consegnati fuori tempo limite non verranno accettati.
- Progetti consegnati senza relazione oppure accompagnati da relazione che non soddisfi le specifiche richieste non saranno ritenuti sufficienti.

9. Divieto di plagio:

- Ogni progetto, elaborato autonomamente e personalmente da ciascuno studente o coppia di studenti, deve essere originale. Codice, o porzione di codice che verrà identificato come "copiato" da altre fonti comporterà l'automatica valutazione insufficiente del progetto. Fanno doverosamente eccezione a questa norma i riferimenti a package e librerie reperite nei repository pubblici per i quali sarà comunque necessario citarne la fonte d'origine in un'apposita sezione bibliografica della relazione.

10. Bibliografia:

Come esempio di partenza per la stesura del progetto si può vedere:

- **OSS Memory Management Simulator** http://www.ontko.com/moss/#memory