Corso di Laurea Triennale in Informatica
Sistemi Operativi
a.a.2013/2014

Progetto Politiche di sostituzione delle pagine (FIFO e LRU)

# 1 Introduzione

La memoria virtuale è una parte della gerarchia di memoria composta dalla memoria e da un disco rigido. Il kernel utilizza la memoria virtuale per ridurre la richiesta di memoria da mettere a disposizione di un processo al fine di poter gestire un maggior numero di processi concorrentemente e poter gestire processi il cui spazio di indirizzamento è maggiore delle dimensioni della memoria a disposizione. La memoria virtuale è implementata attraverso il modello di allocazione non contigua e comprende sia componenti hardware che una componente software (*gestore della memoria virtuale*). Il gestore della memoria virtuale assicura buone prestazioni della memoria virtuale allocando un'adeguata quantità di memoria a un processo utilizzando un algoritmo di sostituzione per rimuovere una porzione che non è stata referenziata di recente.

# 2 Obiettivo

L'obiettivo è l'implementazione di un applicativo che analizzi gli algoritmi di sostituzione delle pagine FIFO (*First-in, First-out*) e LRU (*Least Recently Used*). Supponiamo vi siano $n$ processi e che la memoria venga assegnata in maniera proporzionale alla dimensione del processo. I seguenti parametri devono essere inseriti da linea di comando:

- Numero massimo di frame
- Numero minimo di frame per processo
- Numero di processi
- Dimensione dei processo (ricordiamo che il numero di frame da assegnare è

  $$a_i = \frac{s_i}{\sum s_i} \times m$$

  dove $s_i$ è la dimensione del processo (in pagine) e $m$ sono il numero di frame disponibili
- Lunghezza della successione dei riferimenti

L'applicativo deve simulare i due algoritmi di sostituzione al variare del numero dei frame, supponendo che l'allocazione dei frame sia globale. La gestione dei processi nella coda è scelta liberamente dal candidato.
Il candidato dovrà studiare i page fault in base al numero dei frame, motivando i risultati ottenuti.

# 3 Documentazione

Il progetto deve essere accompagnato dalla seguente documentazione:

1. codice sorgente opportunamente documentato
2. Schema a blocchi degli elementi progettuali del codice: classi, oggetti, funzioni, subroutine e loro connessione logica
3. eventuali file di configurazione dell'applicativo
4. manuale d'uso dell'applicativo
5. elenco delle funzionalità realizzate dall'applicativo
6. documentazione relativa ai risultati ottenuti relativi agli esperimenti che il candidato riterrà opportuno descrivere (correlati di grafici)

Tutta la documentazione, scritta esclusivamente in formato PDF deve essere inserita in una directory (denominata docs), da allegare al progetto stesso. Non saranno prese in considerazione documentazioni pervenute in formati differenti da quelli elencati.

# 4 Linguaggio di programmazione

L'applicativo deve essere sviluppato in Python.

# 5 Avvertenze generali

Il progetto deve essere sviluppato in ambiente Unix/Linux/BSD/Leopard/Lion/Mavericks (a scelta dello studente) e deve essere funzionante in ogni sua parte secondo le specifiche sopra elencate.

# 6 Modalità di presentazione del progetto

Il presente progetto DEVE essere presentato ENTRO 48 ore prima dell'appello di esame (come da calendario del Corso di Laurea) esclusivamente per posta elettronica all'indirizzo sagreste@unime.it avendo l'accortezza di inserire nel campo subject i seguenti dati: progetto PR1 -<numero matricola> - Nome-cognome. Il progetto deve pervenire, sotto forma di allegato, sotto forma di file compresso in modalità .tar.gz. Al suo interno deve trovarsi: l'applicativo, eventuali file di configurazione, la documentazione allegata.

# 7 Bibliografia


- Silberschatz, Galvin, Gagne, *Sistemi Operativi Concetti ed esempi*
- D. M. Dhamdhere, *Sistemi Operativi*
- python web site