**Possibili domande Sistemi Operativi**

**Categoria vero/falso A**

1. In un sistema di memoria a paginazione, il Translation Lookaside Buffer (TLB) velocizza la traduzione di indirizzi virtuali in indirizzi fisici
2. La segmentazione consente a due processi di condividere un segmento
3. La paginazione consente a due processi di condividere una pagina
4. Se non vi sono percorsi chiusi in un grafo di allocazione allora non vi è situazione di stallo
5. FAT è un file system ad allocazione concatenata
6. La politica di scheduling round robin minimizza il tempo medio di attesa dei processi
7. Se un processo è in blocco da 10 ms significa che 10 ms fa ha eseguito una system call
8. Ogni interrupt può essere associato ad un processo che ha richiesto una operazione di I/O
9. Con NTFS è possibile che il file system scriva il contenuto di file di piccola dimensione (es. <1KB) direttamente nell’inode
10. Pwd è un comando GNU/Linux per modificare la password
11. L’i-node contiene al suo interno i dati dei file (es. il testo scritto da un utente in un file .txt)
12. Una system call generata da un processo utente viene gestita in modalità utente
13. Un interrupt viene gestito in modalità utente
14. Il termine copy-on-write indica il caso in cui ad ogni modifica di una variabile in memoria RAM si procede immediatamente con l’aggiornamento anche della sua copia su partizione di disco
15. In un sistema di memoria a paginazione, il Translation Lookaside Buffer (TLB) velocizza la traduzione di indirizzi logici in indirizzi fisici
16. Il meccanismo dei semafori consente forme più generali di sincronizzazione tra processi rispetto alla mutua esclusione
17. In riferimento allo scheduling di processi, la politiche FIFO ed LRU fanno entrambe parte della categoria degli stack algorithms
18. Un processo per creare un nuovo processo deve fare una system call
19. La gestione della memoria con segmentazione può ridurre il consumo di memoria, in quanto consente a più processi di condividere blocchi di codice e di dati
20. Nella gestione della memoria con paginazione, il fenomeno della frammentazione interna è tanto meno rilevante quanto più la lunghezza media dei programmi è grande rispetto alla dimensione della pagina
21. l’attesa indefinita (starvation) è un caso particolare dello stallo
22. In riferimento allo scheduling di processi, la politica FIFO non causa mai meno page fault della politica LRU
23. Lo switch di un processo utente avviene sempre contestualmente a 2 mode switch (utente->kernel, kernel->utente)
24. Un interrupt viene gestito in modalità kernel
25. Il process switch può avvenire sia in modalità kernel che in modalità utente
26. Il termine copy-on-write indica il caso in cui ad ogni modifica di una variabile in memoria RAM si procede immediatamente con l’aggiornamento anche della sua copia su partizione di disco
27. Ogni interrupt può essere associato ad un processo che ha richiesto una operazione di I/O
28. Molti page fault su un processo non modificano le prestazioni degli altri processi

**Categoria risposte multiple B**

1. Un semaforo contatore può:

[A] assumere valori interi arbitrari

[B] assicurare l'accesso in mutua esclusione ad una risorsa condivisa

[C] servire alla realizzazione di una struttura monitor

[D]consentire l'accesso simultaneo di più processi ad una risorsa condivisa, fino al limite fissato dal valore di inizializzazione del contatore

1. Quale tra le seguenti politiche di ordinamento, in generale minimizza il tempo medio di turn-around dei processi:

[A] FCFS

[B] Round-Robin con valutazione dell'attributo di priorità dei processi

[C] Round-Robin senza valutazione dell'attributo di priorità dei processi

[D] Shortest Job First

1. Nella gestione della memoria virtuale tramite paginazione, l’uso di una tabella delle pagine invertita è:

[A] più utile in un sistema con indirizzi a 32 bit rispetto a un sistema con indirizzi a 64 bit

[B] più utile in un sistema con indirizzi a 64 bit rispetto a un sistema con indirizzi a 32 bit

[C] sconsigliabile rispetto all’uso di una tabella delle pagine regolare nei sistemi moderni

[D] sempre sbagliato

1. Data un disco ampio 4 GB, con blocchi ampi 4 KB, e contenente 128 K file, l’ampiezza in B della FAT dipende da:

[A] il numero di file in essa rappresentati

[B] l’ampiezza dei blocchi

[C] l’ampiezza del disco in blocchi e l’ampiezza degli indici di blocco

[D] l’ampiezza del disco

1. La dimensione della Tabella delle Pagine Invertita dipende da:

[A] l’ampiezza in Pagine del disco

[B] l’ampiezza in Pagine della RAM

[C] l’ampiezza in Pagine della Memoria Virtuale

[D] l’ampiezza in Pagine dei file presenti su disco

1. “La dimensione massima di un file ottenibile dal File System dipende dalla contiguità con cui sono scritti i suoi blocchi su disco”. Questa affermazione è vera per:

[A] FAT

[B] ext2

[C] entrambi (sia FAT, sia ext2)

[D] nessuno dei due (né FAT, né ext2)

1. Sia dato un sistema di memoria con indirizzi virtuali suddivisi in 4 campi: a, b, c, d, i primi 3 dei quali siano utilizzati per indirizzare tre livelli gerarchici di tabelle delle pagine e il quarto campo rappresenti l’offset entro la pagina selezionata. Indicare dall’ampiezza di quali campi dipende il numero di pagine indirizzate nel sistema:

[A] da quella di tutti e quattro i campi [B] da quella del campo d

[C] da quella del campo a e d [D] da quelle dei campi a, b, c.

1. Un semaforo binario può:

[A] assumere solo valori pari o discreti

[B] gestire solo l'accesso a due risorse condivise

[C] gestire solo le richieste di accesso provenienti da due processi

[D] assumere solo i valori 0 e 1, con essi denotando “risorsa occupata” e “risorsa libera”.

1. Quale tra le seguenti affermazioni, fatte osservando un grafo di allocazione delle risorse, è certamente vera in generale:

[A] se vi sono percorsi chiusi allora vi è situazione di stallo

[B] se non vi sono percorsi chiusi allora non vi è situazione di stallo

[C] se in un percorso chiuso rilevato si trovano solo risorse a molteplicità unaria, occorre analizzare il caso per decidere

[D] nessuna delle precedenti tre possibili risposte.

1. Quale tra le seguenti affermazioni è corretta in relazione alla politica di ordinamento processi “FCFS senza valutazione dell'attributo di priorità”:

[A] il tempo di attesa è sempre maggiore del tempo di risposta

[B] il tempo di attesa è sempre minore del tempo di risposta

[C] il tempo di attesa è sempre uguale al tempo di risposta

[D] il tempo di attesa ed il tempo di risposta non hanno alcun legame prefissato.

(Si consideri il tempo risposta come il tempo che intercorre dall’attivazione del processo fino alla sua prima esecuzione sul processore)

1. La contiguità dei blocchi in cui viene scritto un file su disco influenza l’overhead generato dal file system:

[A] NTFS

[B] ext2fs

[C] entrambi

[D] nessuno dei due

1. Se un processo è in blocco (in coda di attesa) da 10 ms significa che 10 ms fa ha eseguito:

[A] nessuno dei seguenti

[B] una fork

[C] un context switch

[D] una system call

1. In quale tra i seguenti sistemi operativi è più conveniente l’utilizzo di Inverted Page Tables:

[A] nessuno dei seguenti, il vantaggio è pari per tutti

[B] sistemi a 16 bit

[C] sistemi a 32 bit

[D] sistemi a 64 bit

1. Una system call bloccante causa sempre un context switch:

[A] Sempre

[B] Mai

[C] Sì ma solo se la macchina ha più di un processore

[D] Sì ma solo se c’è qualche altro processo attivo

1. Quale tra i seguenti costituisce un criterio valido di valutazione di una politica di ordinamento di processi:

[A] la capacità di trattare anche processi di lunga durata

[B] il numero di processi completati per unità di tempo

[C] il numero di processi in esecuzione per unità di tempo

[D] il numero di processi in attesa di essere eseguiti.

1. Quale tra le seguenti affermazioni concernenti la politica di ordinamento Round-Robin è corretta:

[A] il tempo di attesa di un processo è sempre maggiore o uguale del suo tempo di risposta

[B] il tempo di attesa di un processo è sempre minore o uguale del suo tempo di risposta

[C] il tempo di attesa di un processo è sempre uguale al suo tempo di risposta

[D] il tempo di attesa di un processo e il suo tempo di risposta non hanno alcun legame prefissato.

1. Data un disco ampio 4 GB, con blocchi ampi 4 KB, e contenente 128 K file, l’ampiezza in B della FAT dipende da:

[A] il numero di file in essa rappresentati

[B] l’ampiezza del disco in blocchi e l’ampiezza degli indici di blocco

[C] l’ampiezza dei blocchi

[D] l’ampiezza del disco

**Categoria risposte aperte C**

1. La dimensione massima di un file ottenibile con file system ext2fs dipende dalla contiguità con cui sono scritti i blocchi del file su disco? Sì / No ?
2. La dimensione massima di un file ottenibile con file system FAT dipende dalla contiguità con cui sono scritti i blocchi del file su disco? Sì / No ?
3. La dimensione massima di un file ottenibile con file system NTFS dipende dalla contiguità con cui sono scritti i blocchi del file su disco? Sì / No ?
4. Elencare le varie politiche di scheduling dei processi. Per ognuna, si discuta in 2-3 righe l’algoritmo alla base del loro funzionamento. (Es. “Scheduling a Lotteria: il prossimo processo ad andare in esecuzione è scelto a caso, con un generatore di numeri random, tra quelli presenti in coda di pronti e una volta in esecuzione vi rimane fino alla sua terminazione, senza poter essere prerilasciato”).
5. Considerando le politiche di ordinamento FCFS, RR e SRTN è possibile dire che qualcuna di esse tende a ridurre o addirittura minimizzare, rispetto alle altre politiche, qualcuna delle seguenti metriche: tempo medio di risposta, tempo medio di attesa, tempo medio di turn around? (Specificare e Commentare anche tenendo conto del fatto che “ridurre” e “minimizzare” sono termini con un significato ben preciso).
6. Si descriva il funzionamento delle seguenti tre politiche di rimpiazzo: optimal, NRU (Not Recently Used), aging. Possono essere usati anche diagrammi descrittivi.
7. Si consideri un sistema che utilizza la paginazione per gestire la memoria. Si discutano brevemente vantaggi e svantaggi nell’adottare pagine di dimensione ampia oppure di piccola.

[A] Pagine di dimensione ampia (vantaggi e svantaggi)

[B] Pagine di dimensione piccola (vantaggi e svantaggi)

Come visto in classe, il valore ottimo di dimensione di una pagina può essere definito matematicamente. Utilizzando i seguenti parametri:

– σ byte dimensione media di un processo

– π byte dimensione media di una pagina

– ε byte per riga in tabella delle pagine

[C] si scriva innanzitutto una funzione f(π) che definisca matematicamente lo spreco di memoria in maniera dipendente dalla dimensione π di una pagina.

– f(π) =

[D] si determini quindi il valore ottimo di π che minimizza lo spreco di memoria.

1. Si consideri la politica di scheduling Round Robin di quanto q e si supponga che in un sistema ci siano N processi interattivi tutti con lo stesso comportamento. Ciascuna interazione dà luogo ad un CPU burst che richiede la CPU per un tempo c.

[A] Se c < q quanto tempo aspetta al più un processo in coda ready prima di ottenere la CPU?

[B] Se c < q quanto tempo aspetta al più l'utente prima che la CPU finisca di elaborare l'interazione?

[C] Se c > q quanto tempo aspetta l'utente prima di iniziare l’ultimo quanto di interazione per l’ultimo processo rimasto ancora attivo? (Si consideri che c può essere espresso come c = aq + b; ovvero a quanti di tempo + b<c tempo nell’ultimo quanto)

1. E’ noto che esistano 4 condizioni che concorrono all’insorgere di situazioni di stallo. Lo studente le elenchi illustrandole brevemente, ovvero in non più di due/tre righe ciascuna.
2. Illustrare con un diagramma come quello visto a lezione gli stati in cui può trovarsi un processo e le transizioni tra essi. (Presentare solo il diagramma/figura, con i nomi degli stati e delle transizioni; non sono necessarie ulteriori spiegazioni).
3. Ogni file è caratterizzato da vari attributi che, direttamente o dopo l’esecuzione di semplici comandi, sono anche visibili agli utenti. Si elenchino i principali tra gli attributi di file
4. La dimensione max di un file ottenibile con ext2fs dipende dalla contiguità con cui sono scritti i blocchi del file su disco?
5. La dimensione max di un file ottenibile con FAT dipende dalla contiguità con cui sono scritti i blocchi del file su disco?
6. La dimensione max di un file ottenibile con NTFS dipende dalla contiguità con cui sono scritti i blocchi del file su disco?
7. In quale tra i seguenti sistemi operativi è più conveniente l’utilizzo di Inverted Page Tables:

1. nessuno dei seguenti, il vantaggio è pari per tutti

2. sistemi a 16 bit

3. sistemi a 32 bit

4. sistemi a 64 bit

Motivare BREVEMENTE la risposta.

1. Si consideri il caso di un file system NTFS; si discuta brevemente (anche con uso di diagrammi) se e perché la struttura di overhead per memorizzare un file risenta della contiguità con cui i blocchi di tale file sono disposti sul disco fisso.
2. Si consideri il caso di un file system ext2; si discuta brevemente (anche con uso di diagrammi) se e perché la struttura di overhead per memorizzare un file risenta della contiguità con cui i blocchi di tale file sono disposti sul disco fisso.
3. Gli hard disk sono componenti molto importanti di un computer che permettono di immagazzinare permanentemente un insieme moderatamente grande di informazioni. Il sistema operativo si occupa di gestire anche queste componenti hardware permettendo, ad esempio, operazioni su file quali memorizzazione, recupero, cancellazione, ecc. I computer moderni sono dotati di hard disk di capacità sempre maggiore fornendo dunque un vantaggio in termini di spazio di memorizzazione agli utenti ma anche nuove complessità di gestione per il sistema operativo. Lo studente illustri, in massimo una pagina, le implicazioni (es. problematiche e possibili soluzioni, ma anche semplificazioni che diventerebbero possibili) per le varie componenti e strutture di un sistema operativo che si trovasse a dover gestire un hard disk di capacità infinita.
4. Si consideri un sistema con un solo processo attivo, il quale sta eseguendo la seguente porzione di codice:

pid\_1 = fork ();

pid\_2 = fork ();

pid\_3 = fork ();

Quanti processi risulteranno attivi alla fine dell’esecuzione di queste tre line di codice? (Si motivi la risposta)

1. Si consideri l'algoritmo AGING di page replacement con contatore (o stimatore) di 3 bit, e una memoria di 4 frames contenenti rispettivamente le pagine 1 2 3 e 4 di un certo processo. Si supponga che subito dopo uno sweep (aggiornamento del contatore) all’istante t0 i contatori siano inizializzati come segue:

contatore pagina 1: 110

contatore pagina 2: 111

contatore pagina 3: 101

contatore pagina 4: 100

All'istante t1 avviene uno sweep. Tra t0 e t1 è stata eseguita la seguente sequenza di accessi a memoria, nell’ordine:

pagina 4, pagina 2; pagina 4, pagina 2.

[A] Che valore avranno i contatori dopo lo sweep in t1?

contatore pagina 1: \_\_\_\_ contatore pagina 2: \_\_\_\_

contatorepagina 3: \_\_\_\_ contatore pagina 4: \_\_\_\_

[B] Supponendo invece che subito dopo t0 fosse avvenuto un page fault, di quale pagina avrebbe causato la sostituzione? Perché?

1. Si elenchino, senza spiegarle, le condizioni relative al verificarsi dello stallo (deadlock).
2. Si consideri l'algoritmo AGING di page replacement con contatore (o stimatore) di 3 bit, e una memoria di 4 frames contenenti rispettivamente le pagine 0 1 2 e 3 di un certo processo. Si supponga che subito dopo uno sweep (aggiornamento del contatore) all’istante t0 i contatori siano inizializzati come segue:

contatore pagina 0: 110

contatore pagina 1: 111

contatore pagina 2: 001

contatore pagina 3: 100

All'istante t1 avviene uno sweep.

Tra t0 e t1 è stata eseguita la seguente sequenza di accessi a memoria, nell’ordine:

pagina 4, pagina 2; pagina 4, pagina 2.

[A] Che valore assumeranno i contatori considerando lo sweep in t1 e i seguenti a memoria: pag 2, pag 1, pag 1?

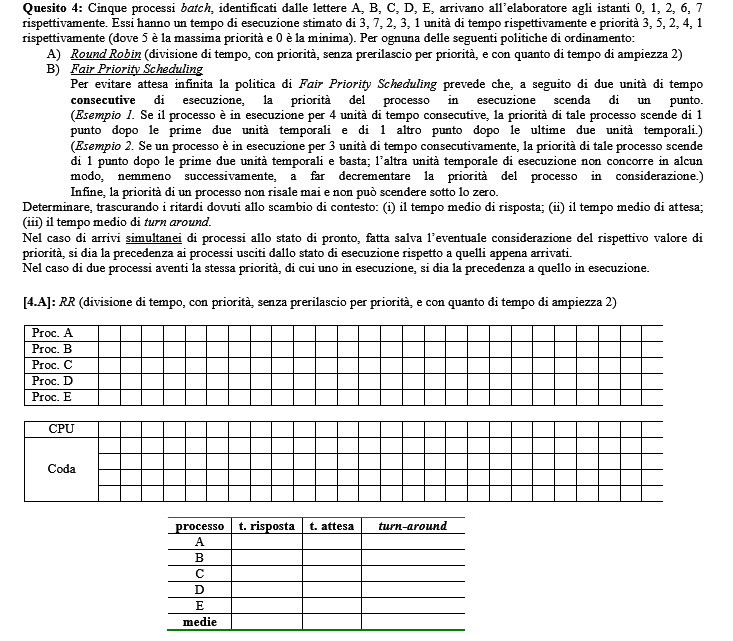
contatore pagina 0: \_\_\_\_ contatore pagina 1: \_\_\_\_

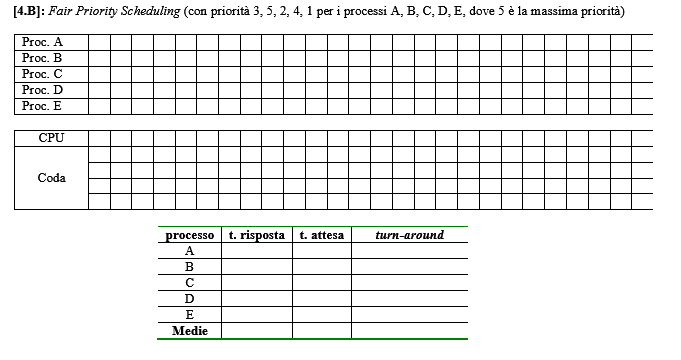
contatorepagina 2: \_\_\_\_ contatore pagina 3: \_\_\_\_

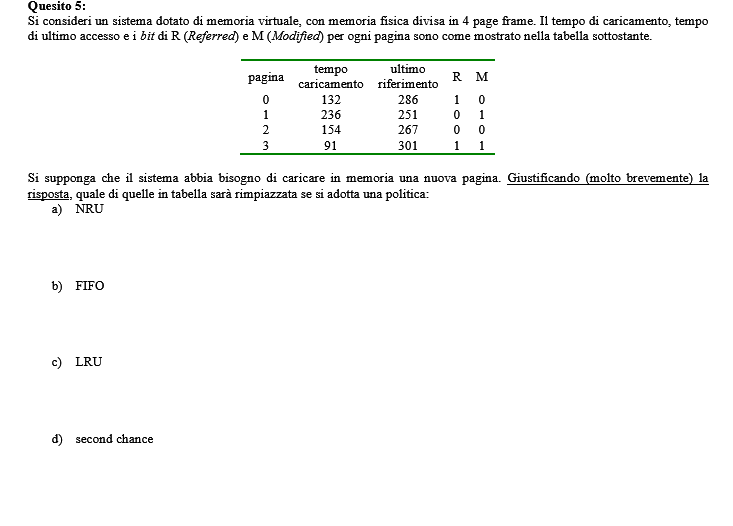
[B] Supponendo invece che subito dopo t0 fosse avvenuto un page fault, di quale pagina avrebbe causato la sostituzione? Perché?

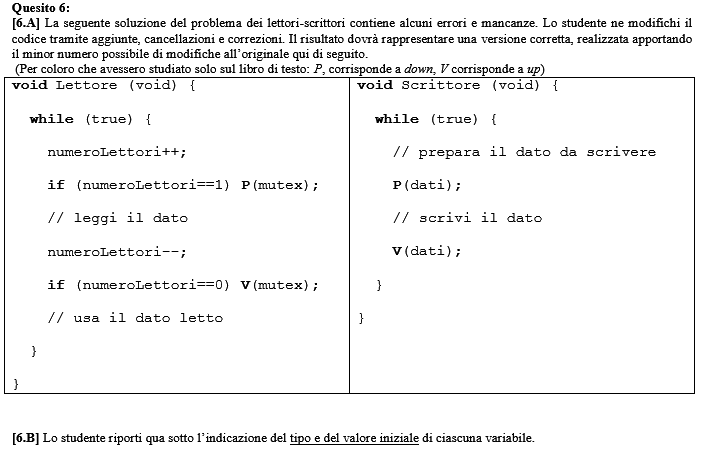
**Categoria esercizi D**

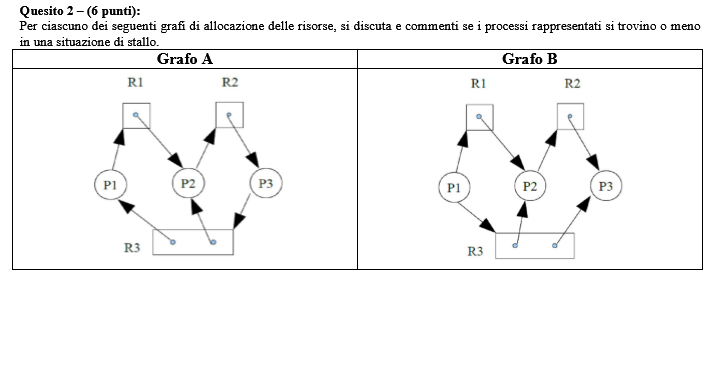
1. Sia data una partizione di disco ampia 64 GB organizzata in blocchi dati di ampiezza 1 kB. In caso serva, si consideri l’ipotesi di contiguità nulla di un file (ciascun blocco si trova su disco in posizione non adiacente al blocco precedente e a quello successivo nella composizione del file). Si determini l’ampiezza massima di file ottenibile per l’architettura di file system ext2fs assumendo i-node ampi 128 B, i-node principale contenente 12 indici di blocco e 1 indice di I, II e III indirezione ciascuno. Si determini poi il rapporto inflattivo che ne risulta, ossia l’onere proporzionale dovuto alla memorizzazione della struttura di rappresentazione rispetto a quella dei dati veri e propri.





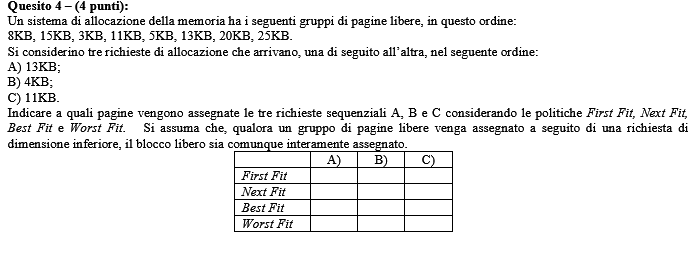


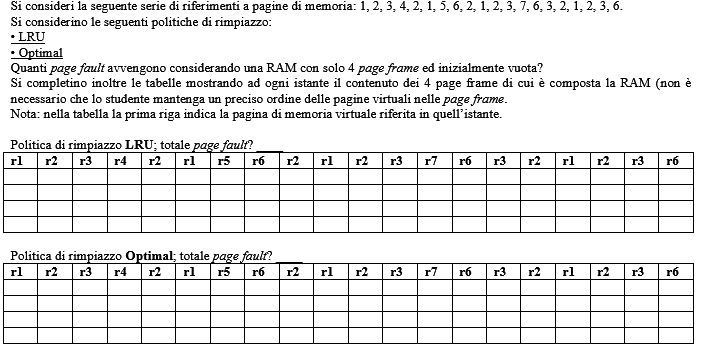


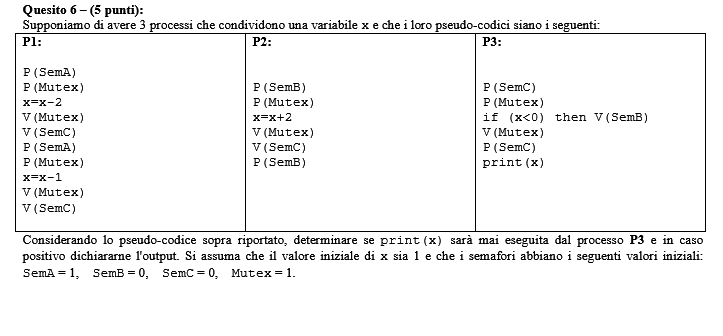


1. Sia data una partizione di disco ampia 256 GB organizzata in blocchi dati di ampiezza 1 KB. Sotto queste ipotesi e assumendo che gli indirizzi debbano essere espressi con un numero di bit multiplo di 8, si determini l’ampiezza massima di file ottenibile per l’architettura di file system ext2fs nel caso pessimo di contiguità nulla, assumendo i-node ampi 512 B, i-node principale contenente 13 indici di blocco e 1 indice di I, II e III indirezione ciascuno.

Si determini poi il rapporto inflattivo che ne risulta, ossia l’onere proporzionale dovuto alla memorizzazione delle strutture di rappresentazione rispetto a quella dei dati veri e propri. Effettuati tali calcoli si discuta se e con quale rapporto inflattivo l’architettura FAT e NTFS possano rappresentare file di tale ampiezza nella partizione data, sotto le medesime ipotesi di contiguità nulla. Per l’architettura NTFS si assumano record ampi 512 B, 208 B riservati all’attributo dati nel record principale e 400 B nei record di estensione.

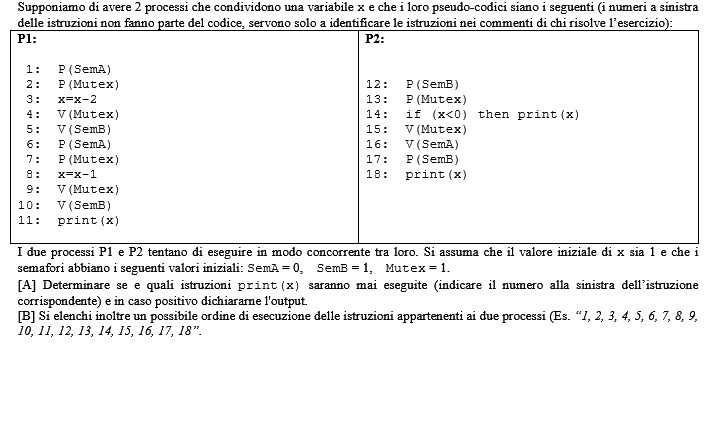






Si assuma che il valore iniziale di x sia 0 e che i tre semafori abbiano i seguenti valori iniziali: SemA = 1, SemB = 0, SemC = 0, Mutex = 1.

Si discuta l’ordine di esecuzione dei vari processi (e delle loro istruzioni) e si determini l'output print(x) del processo P3



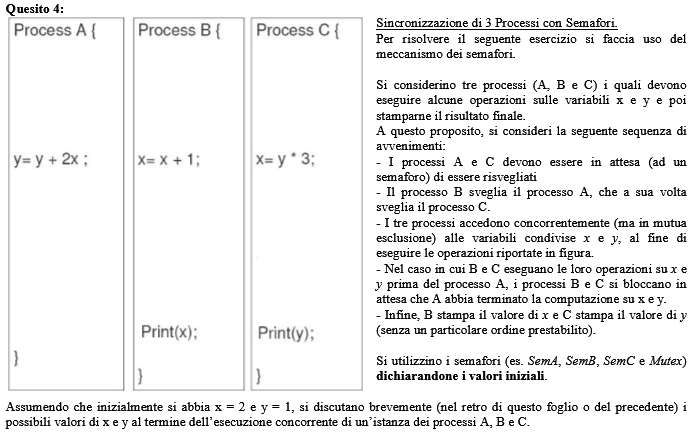
1. Sia data una partizione di disco ampia 128 GB organizzata in blocchi dati di ampiezza 1 KB e indici di dimensione 32 bit. Il sistema operativo utilizzato adotta un’architettura di file system di tipo ext2fs con i-node ampi esattamente un blocco, i-node principale contenente 12 indici di blocco, 1 indice di I indirezione e 1 indice di II indirezione. In caso serva, si consideri l’ipotesi di contiguità del file (ciascun blocco si trova su disco in posizione adiacente al blocco precedente e a quello successivo nella composizione del file).

[A] Vi sono delle informazioni nel testo che non sono necessarie/utili per risolvere l’esercizio? Se sì spiegare quali sono e perché non sono necessarie/utili.

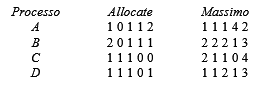
[B] Si determini l’ampiezza massima di file ottenibile per l’architettura di file system ext2fs sopra descritta.

[C] Si determini la quantità di spazio occupata dalla struttura di i-node necessaria a rappresentare un tale file.

[D] Quanto spazio sarebbe invece occupato dalla struttura di i-node per rappresentare un file contenente 1MB di dati?



1. Un sistema ha 4 processi e 5 risorse da ripartire. L’attuale allocazione e i bisogni massimi sono i seguenti:

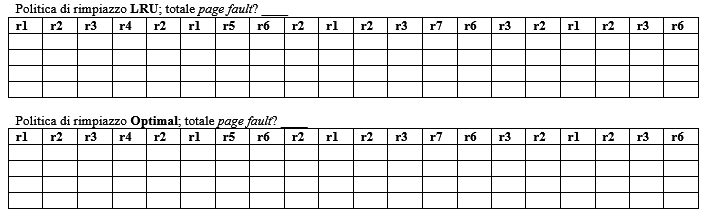


[A] Considerando il vettore delle risorse disponibili uguale a [0 0 1 2 x], si discuta per quale valore minimo di x questo sia uno stato sicuro e quando invece sia a rischio di deadlock.

[B] Per risolvere l’esercizio lo studente ha di fatto ripetutamente utilizzato una parte di un noto

algoritmo. Tale algoritmo assegna risorse a processi solo se l’assegnazione fa rimanere il sistema in uno stato sicuro. Come si chiama questo algoritmo?

1. Si consideri la seguente serie di riferimenti a pagine di memoria: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. Si considerino le seguenti politiche di rimpiazzo LRU ed Optimal. Quanti page fault avvengono considerando una RAM con solo 4 page frame ed inizialmente vuota? Si completino inoltre le tabelle mostrando ad ogni istante il contenuto dei 4 page frame di cui è composta la RAM (non è necessario che lo studente mantenga un preciso ordine delle pagine virtuali nelle page frame).



1. Sia data una partizione di disco ampia 64 GB organizzata in blocchi dati di ampiezza 1 KB e indici di dimensione 32 bit. In caso serva, si consideri l’ipotesi di contiguità nulla di un file (ciascun blocco si trova su disco in posizione non adiacente al blocco precedente e a quello successivo nella composizione del file).

[A] Si determini l’ampiezza massima di file ottenibile per l’architettura di file system ext2fs assumendo i-node ampi esattamente un blocco, i-node principale contenente 12 indici di blocco, 1 indice di I indirezione e 1 indice di II indirezione.

[B] Si determini la quantità di spazio occupata dalla struttura di i-node necessaria a rappresentare un tale file.

[C] Si determini la quantità di spazio occupata dal totale della struttura FAT in caso il file system usato per rappresentare il file calcolato in [4.A] nel sistema sopra descritto sia basato su FAT invece che su i-node (ext2fs).

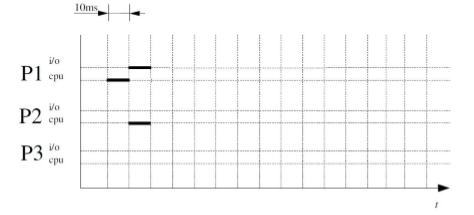
1. Si considerino tre processi che alternano fasi di uso di una CPU (job) con fasi di I/O burst. Si utilizzi una politica di CPU scheduling del tipo shortest remaining time next (SRTN) applicata sui vari job pronti per l’esecuzione sulla CPU. I tre processi, denominati P1, P2 e P3, sono caratterizzati dalle seguenti sequenze di CPU burst e I/O burst.

P1: cpu-10ms, i/o-10ms, cpu-30ms, i/o-10ms, cpu-10ms.

P2: cpu-20ms, i/o-10ms, cpu-10ms.

P3: cpu-50ms, i/o-10ms, cpu-10ms.

Ovviamente nessun processo può avanzare alla fase successiva senza aver prima completato le precedenti, nell’ordine. Si supponga che i tre processi facciano I/O su dispositivi distinti. Si mostri, in ciascun istante di tempo, quali processi risultano in stato di running sulla CPU e quali in blocco I/O, marcando, nel diagramma sottostante, le linee di CPU o di I/O corrispondenti a ciascun processo.

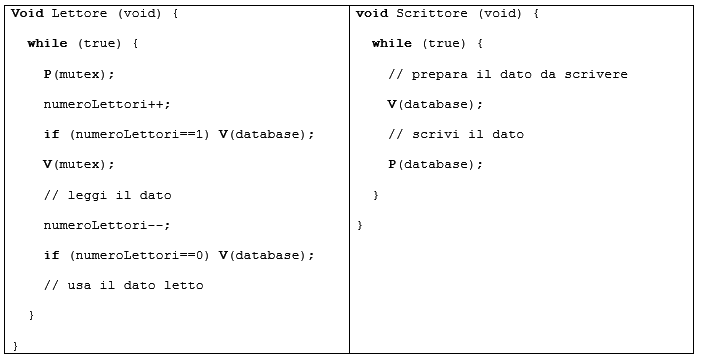


1. Il problema del “produttore/consumatore” è un classico problema di sincronizzazione tra più processi che accedono concorrentemente a risorse condivise.

[A] Lo studente utilizzi i monitor per scrivere due procedure chiamate Producer e Consumer che possano essere eseguite concorrentemente al fine di risolvere il problema evitando il deadlock del sistema. (Si consideri il caso in cui le risorse prodotte e non ancora consumate possano essere al massimo N).

[B] Lo studente utilizzi i semafori per risolvere lo stesso problema

1. La seguente soluzione del problema dei lettori-scrittori contiene cinque banali errori o mancanze. Lo studente ne modifichi il codice tramite aggiunte, cancellazioni e correzioni. Il risultato dovrà rappresentare una versione corretta, realizzata apportando il minor numero possibile di modifiche all’originale qui di seguito. (Per coloro che avessero studiato solo sul libro di testo: P, corrisponde a down, V corrisponde a up) Inoltre, lo studente dichiari i valori iniziali delle variabili numeroLettori, mutex e database.



1. Sia data una partizione di disco ampia 64 MB organizzata in blocchi dati di ampiezza 1 kB. In caso serva, si consideri l’ipotesi di contiguità nulla di un file (ciascun blocco si trova su disco in posizione non adiacente al blocco precedente e a quello successivo nella composizione del file). Si consideri che su tale partizione è in uso un file system di tipo ext2 con i-node ampi un blocco e si assuma che l’i-node principale contenga 10 indici di blocco e 1 indice di I e II indirezione ciascuno. Considerando che la dimensione di un file include sia i Byte dei blocchi di dati, sia i Byte occupati dagli i-node necessari a indirizzare i blocchi di dati, si determini l’ampiezza massima di un file consentita dal sistema.
2. I “filosofi a cena” è un classico problema di sincronizzazione tra più processi.

[A] Lo studente descriva brevemente di che problema si tratta e le analogie con scenari inerenti i sistemi operativi.

[B] Come visto in aula, lo studente utilizzi i monitor per scrivere una procedura Filosofo che possa essere eseguita concorrentemente ad altre (come fossero un gruppo di filosofi a cena, appunto) evitando deadlock e starvation. A questo proposito si consideri un tavolo con N filosofi ed N forchette.

1. In una chiavetta USB da 2 GB con filesystem FAT32 e blocchi da 4KB vengono registrati 100 files da 1000 blocchi ciascuno.

Considerando che:

il tempo di lettura di un blocco è 100 µs;

il tempo medio di accesso (tempo per raggiungere qualsiasi blocco da un altro qualsiasi, purché non contiguo altrimenti il tempo è nullo) è di 5 ms;

si presuma che la FAT sia precaricata in memoria (si trascurino pertanto i tempi di accesso alla stessa).

Calcolare il tempo necessario a leggere tutti i files, nelle ipotesi

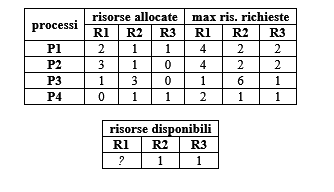
[A] di minima frammentazione;

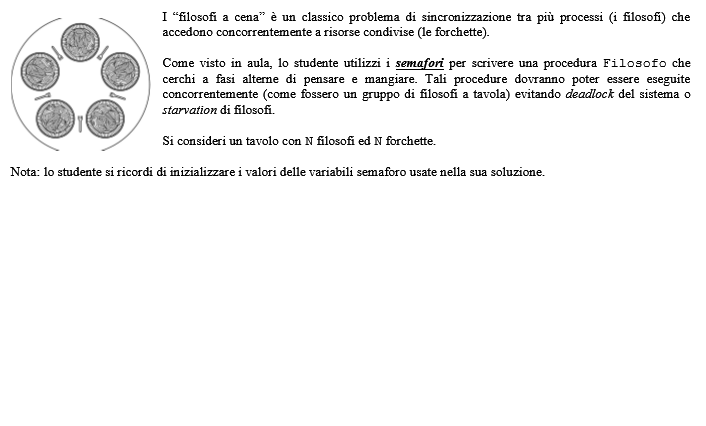
[B] di massima frammentazione;

Si calcoli inoltre

[C] il tempo di scanning dello spazio libero, nel caso [A]

1. In aula si è discusso di come l’algoritmo del banchiere faccia procedere il sistema attraverso una serie di stati sicuri (safe). Data la situazione schematizzata nelle tabelle seguenti, quante istanze della risorsa R1 devono essere disponibili affinché il sistema sia safe? (Commentare la scelta)



1. 

Si ripeta l’esercizio sui “filosofi a cena” risolvendolo però con i monitor

1. Sia data una partizione di disco ampia 256 GB organizzata in blocchi dati di ampiezza 1 KB. Sotto queste ipotesi si calcoli la dimensione minima in bit di un indirizzo considerando il vincolo che la dimensione di un indirizzo debba essere un multiplo di 8 bit. Si determini conseguentemente l’ampiezza massima di file ottenibile per l’architettura di file system ext2fs nel caso pessimo di contiguità nulla, assumendo i-node ampi come un blocco, i-node principale contenente 13 indici di blocco e 1 indice di I, II e III indirezione ciascuno. Si determini poi il rapporto inflattivo che ne risulta, ossia l’onere proporzionale dovuto alla memorizzazione delle strutture di rappresentazione rispetto a quella dei dati veri e propri. Effettuati tali calcoli si discuta se e con quale rapporto inflattivo le architetture FAT possa rappresentare file di tale ampiezza nella partizione data, sotto le medesime ipotesi di contiguità nulla.
2. Si consideri un sistema composto da quattro processi (P1, P2, P3, P4), e quattro tipologie di risorse (R1, R2, R3, R4) con disponibilità: 1 risorsa di tipo R1, 1 risorsa di tipo R2, 1 risorsa di tipo R3, 2 risorse di tipo R4. Si assuma che:

- ogni volta che un processo richieda una risorsa libera, questa venga assegnata al processo richiedente;

- ogni volta che un processo richieda una risorsa già occupata, il processo richiedente deve attendere che la risorsa si liberi prima di potersene impossessare (utilizzando una coda FIFO di processi in attesa di una determinata risorsa)

Si consideri la seguente successione cronologica di richieste e rilasci di risorse:

1) P2 richiede R1,R2,R3

2) P3 richiede R2,R4

3) P2 rilascia R2

4) P4 richiede R4

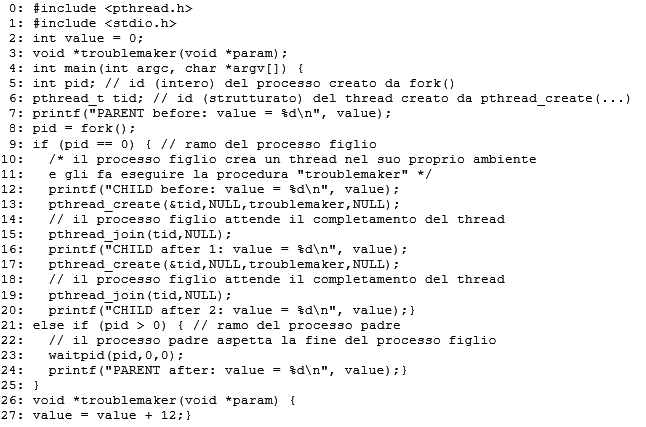
5) P1 richiede R1

6) P2 richiede R2

7) P3 richiede R3

Verificare se alla fine di questa serie di operazioni il sistema si trovi in condizioni di stallo (suggerimento: usare un grafo di allocazione delle risorse)

1. Il programma in linguaggio C riportato sotto, utilizzabile in ambiente GNU/Linux, mostra come un processo utente possa creare sia un processo figlio (tramite la chiamata fork() alla linea 8) che un thread (tramite la chiamata pthread create(...) alle linee 13 e 17). I flussi di controllo risultanti dall’esecuzione del programma condividono la variabile value inizializzata a 0 alla linea 2. Lo studente indichi quale valore tale variabile avrà quando sarà stampato alle linee 7, 12, 16, 20, 24 illustrando i meccanismi di livello di sistema operativo che intervengono per produrre tale effetto.



1. Il problema del “produttore/consumatore” è un classico problema di sincronizzazione tra più processi che accedono concorrentemente a risorse condivise. Lo studente descriva concisamente tale problema.

Inoltre, lo studente utilizzi i monitor per scrivere due procedure chiamate Producer e Consumer che possano essere eseguite concorrentemente al fine di risolvere il problema evitando il deadlock del sistema. (Si consideri il caso in cui le risorse prodotte e non ancora consumate possano essere al massimo N).

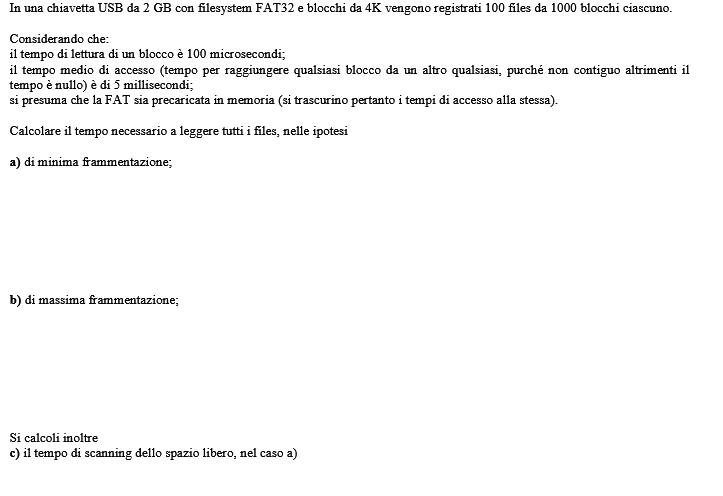
1. Una "chiavetta USB" da 8 GB è formattata con un filesystem di tipo FAT con blocchi da 4kB.

[A]Calcolare la dimensione di ogni record della FAT (scegliendo fra lunghezze che siano potenze di 2; ovvero di 8, 16, 32 e 64 bit, non altri valori intermedi) e la dimensione totale della FAT.

[B]

1. Sia data una partizione di disco ampia 64 GB organizzata in blocchi dati di ampiezza 1 KB. In caso serva, si consideri l’ipotesi di contiguità nulla di un file (ciascun blocco si trova su disco in posizione non adiacente al blocco precedente e a quello successivo nella composizione del file).

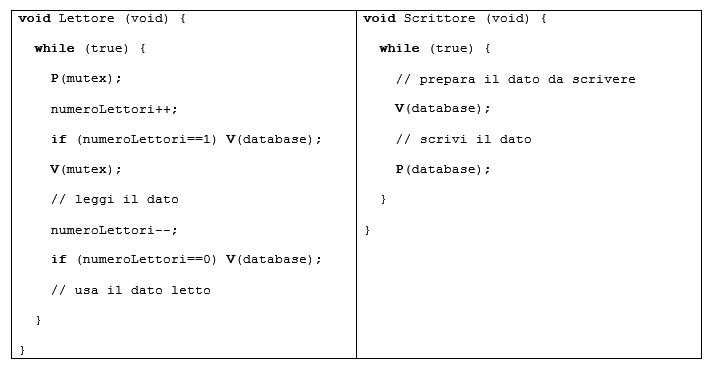
Si determini l’ampiezza massima di file ottenibile per l’architettura di file system ext2fs assumendo i-node ampi esattamente un blocco, i-node principale contenente 12 indici di blocco, 1 indice di I indirezione e 1 indice di II indirezione. Si determini inoltre la quantità di spazio occupata dalla struttura di i-node necessaria a rappresentare un tale file.

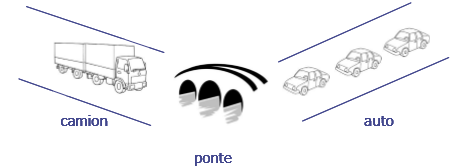
1. 
2. Un sistema ha 4 processi e 5 risorse da ripartire. L’attuale allocazione e i bisogni massimi sono i seguenti:



[A] Considerando il vettore delle risorse disponibili uguale a [0 0 x 1 2], si discuta per quale valore minimo di x questo sia uno stato sicuro e quando invece sia a rischio di deadlock.

[B] Per risolvere l’esercizio lo studente ha di fatto ripetutamente utilizzato una parte di un noto algoritmo. Tale algoritmo assegna risorse a processi solo se l’assegnazione fa rimanere il sistema in uno stato sicuro. Come si chiama questo algoritmo?

1. Sia data una partizione di disco ampia 64 GB organizzata in blocchi dati di ampiezza 1 kB. In caso serva, si consideri l’ipotesi di contiguità nulla di un file (ciascun blocco si trova su disco in posizione non adiacente al blocco precedente e a quello successivo nella composizione del file). Si determini l’ampiezza massima di file ottenibile per l’architettura di file system ext2fs assumendo i-node ampi 128 B, i-node principale contenente 12 indici di blocco e 1 indice di I, II e III indirezione ciascuno. Si determini poi il rapporto inflattivo che ne risulta, ossia l’onere proporzionale dovuto alla memorizzazione della struttura di rappresentazione rispetto a quella dei dati veri e propri.
2. La seguente soluzione del problema dei lettori-scrittori contiene alcuni errori e mancanze. Lo studente ne modifichi il codice tramite aggiunte, cancellazioni e correzioni. Il risultato dovrà rappresentare una versione corretta, realizzata apportando il minor numero possibile di modifiche all’originale qui di seguito. (Per coloro che avessero studiato solo sul libro di testo: P, corrisponde a down, V corrisponde a up) 
3. Si consideri la situazione rappresentata nella figura sottostante dove due strade sono unite da un ponte. Tale ponte è troppo stretto per avere due sensi di marcia: il ponte è dunque a senso unico alternato. Non c’è precedenza prefissata, chi arriva prima comincia, se può, ad attraversare il ponte e fintanto che il ponte non si libera completamente non si può invertire senso di marcia sul ponte. Si assuma che da un lato del ponte arrivino (per attraversare il ponte) solo camion mentre dall’altro lato arrivino solo auto. Il ponte è sufficientemente robusto da sopportare il passaggio di quante auto si vuole contemporaneamente, ma può sopportare solo un camion alla volta (se un camion sta attraversando il ponte sarà l’unico mezzo sul ponte.)



Si scrivano dei processi che rappresentino i “camion” e dei processi che rappresentino delle “auto” mentre cercano di accedere alla risorsa condivisa “ponte” rispettando le condizioni sopra descritte e utilizzando i semafori in modo da sincronizzare il sistema senza incorrere in deadlock. Nota: Lo studente si ricordi di inizializzare i valori delle variabili semaforo usate nella sua soluzione

1. Il problema dei lettori e scrittori è un classico problema di sincronizzazione tra più processi che accedono concorrentemente a una risorsa condivisa. A) Si descriva e discuta molto brevemente questo problema. B) Inoltre, utilizzando i semafori si proponga una procedura reader e una procedura writer in grado di rappresentarne una soluzione.
2. Un sistema di controllo di una cisterna misura la quantità d'acqua contenuta nella stessa misurandone l'altezza ogni secondo facendo uso di un apposito sensore. Grazie a questa misura una centralina prende le dovute decisioni di svuotamento o riempimento. Il sensore comunica con la centralina per mezzo di una linea seriale asincrona; il sensore invia ogni misura corredandola di alcuni dati come segue:

2 byte per il proprio identificativo univoco;

4 byte per contenere un timestamp;

2 byte per la misura vera e propria;

1 byte contenente un dato ausiliario (crc).

Assumendo che:

la misura viene effettuata e spedita alla centralina ogni secondo; l'invio consiste nella spedizione da parte del sensore alla centralina del pacchetto di dati come sopra descritto; la linea seriale asincrona è configurata per funzionare ad una velocità di 1200 baud, "8N1" (8 bit di dati, nessuna parità, un solo bit di stop);

[A] Calcolare la percentuale di utilizzo della linea (tale percentuale è il rapporto fra la banda realmente usata e la velocità massima permessa).

Assumendo invece che:

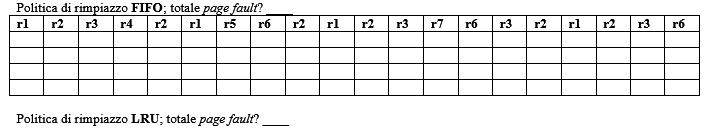
la centralina deve loggare le misure su una memoria persistente per 30 giorni; la centralina bufferizza i dati ricevuti e li scrive tal quali (=in formato grezzo) all'interno di un file binario, registrandolo su base oraria (ogni ora cambia file); i files vengono registrati su un filesystem basato su i-node, blocchi da 1KB, assumendo i-node ampi 128 B, record da 32 bit, inode principale contenente 12 indici di blocco e 1 indice di I, II e III indirezione ciascuno.

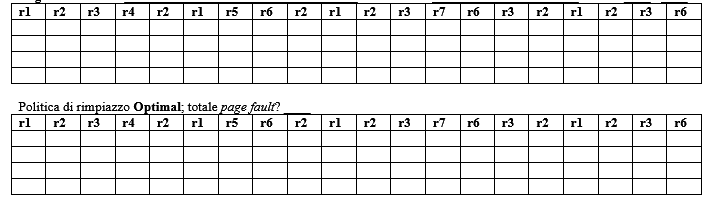
[B] Calcolare il rapporto inflattivo nel caso in cui i files siano lasciati come files singoli contenenti i dati orari (pertanto in tutto: 24 x 30 = 720 files)

1. Si consideri la seguente serie di riferimenti a pagine di memoria: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. Si considerino le seguenti politiche di rimpiazzo:

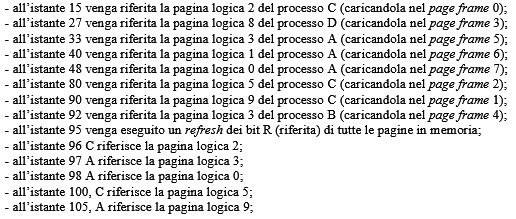
* FIFO
* LRU
* Optimal

Quanti page fault avvengono considerando una RAM con solo 4 page frame ed inizialmente vuota? Si completino inoltre le tabelle mostrando ad ogni istante il contenuto dei 4 page frame di cui è composta la RAM (non è necessario che lo studente mantenga un preciso ordine delle pagine virtuali nelle page frame. Nota: nella tabella la prima riga indica la pagina di memoria virtuale riferita in quell’istante.





1. Si consideri un sistema dotato di memoria virtuale, con memoria fisica (RAM) divisa in 8 page frame condivisa da 4 processi contemporaneamente attivi: A, B, C e D. Si supponga che la RAM sia inizialmente vuota e che ogni riferimento a una pagina non presente in RAM comporti il suo caricamento in RAM (se serve, con rimpiazzo di un’altra pagina già presente). Si supponga inoltre che:



Se utile, si assuma che nessuna pagina sia modificata durante tutta l’esecuzione. Si supponga che il sistema utilizzi un algoritmo di sostituzione second chance (locale) Si supponga inoltre che a supporto di tale algoritmo vi sia una o più liste ordinate delle pagine accedute ad un dato istante e che tale/tali lista/liste siano composti di elementi del tipo: C2(15, 1) dove C2 indica che si tratta della pagina logica 2 del processo C, mentre (15, 1) indica che tale pagina è stata caricata in memoria all’istante 15 e che il suo bit ‘riferita’ è uguale a 1.

Assumendo che non accada più nulla di significativo fino all’istante 110 si scriva/scrivano:

[A] la/le lista/liste delle pagine all’istante 85

[B] la/le lista/liste delle pagine all’istante 101

[C] la/le lista/liste delle pagine all’istante 106