

DOMANDA	Vero/Falso
In un sistema di memoria a paginazione, il <i>Translation Lookaside Buffer</i> (TLB) velocizza la traduzione di indirizzi fisici in indirizzi logici	F
La gestione della memoria con segmentazione consente a più processi di condividere segmenti contenenti codice o dati	V
<code>chmod</code> è un comando utilizzabile nei sistemi GNU/Linux per modificare i permessi di file e directory	V
GNU/Linux tende a scrivere i file su disco come blocchi sequenziali contigui	V
Con NTFS è possibile che il file system scriva il contenuto (i dati) di file di piccola dimensione (es. <1KB) direttamente dentro l'i-node principale	V
<code>rmdir</code> è un comando POSIX per modificare il nome di una directory	F

DOMANDA	Vero/Falso
La tecnica di allocazione contigua è soggetta al problema della frammentazione esterna	V
Un indirizzo logico identifica direttamente una cella di memoria fisica	F
Il <i>Process Control Block</i> (PCB) è una struttura dati che memorizza tutte le informazioni riguardanti un processo	V
Le chiamate di sistema comportano il cambiamento della modalità esecuzione da spazio kernel verso spazio utente	F
La tecnica di compattazione della memoria non comporta la rilocalizzazione dei programmi	F
Il principio di località dei riferimenti è la motivazione base che rende efficace l'uso di cache	V
La tecnica di paginazione è soggetta alla frammentazione interna ma non a quella esterna	V
La politica di scheduling <i>Shortest Job First</i> (SJF) non è soggetta al fenomeno di <i>starvation</i>	F
Un <i>page fault</i> avviene quando la entry richiesta non è presente nel <i>Translation Lookaside Buffer</i> (TLB)	F
In un sistema combinato di paginazione/segmentazione lo spazio di indirizzamento utente è suddiviso in un certo numero di pagine a dimensione fissa, che a loro turno sono suddivise in un certo numero di segmenti	F

DOMANDA	Vero/Falso
In un sistema di memoria a paginazione, il <i>Translation Lookaside Buffer</i> (TLB) velocizza la traduzione di indirizzi fisici in indirizzi virtuali	F
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	V
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	V
Il meccanismo dei semafori consente forme più generali di sincronizzazione tra processi rispetto alla mutua esclusione	V
L'attesa indefinita (<i>starvation</i>) è un caso particolare dello stallo	F
In riferimento allo <i>scheduling</i> di processi, la politica FIFO non causa mai meno <i>page fault</i> della politica LRU	F
Una system call generata da un processo utente viene gestita in modalità utente	F
Lo <i>switch</i> di un processo utente avviene sempre contestualmente a 2 <i>mode switch</i> (utente->kernel, kernel->utente)	V
Un interrupt viene gestito in modalità <i>kernel</i>	V
Il process switch può avvenire sia in modalità kernel che in modalità utente	F
Il termine <i>copy-on-write</i> 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	F
Ogni interrupt può essere associato ad un processo che ha richiesto una operazione di I/O	F
Molti <i>page fault</i> su un processo non modificano le prestazioni degli altri processi	F
Un processo per lanciare un nuovo processo deve fare una system call	V

DOMANDA	Vero/Falso
L'i-node contiene al suo interno i dati dei file (es. il testo scritto da un utente in un file .txt)	F
Una system call generata da un processo utente viene gestita in modalità utente	F
Un <i>interrupt</i> viene gestito in modalità utente	F
Il termine <i>copy-on-write</i> 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	Più F che V ma entrambe ok
In un sistema di memoria a paginazione, il <i>Translation Lookaside Buffer</i> (TLB) velocizza la traduzione di indirizzi virtuali in indirizzi fisici	V
Il meccanismo dei semafori consente forme più generali di sincronizzazione tra processi rispetto alla mutua esclusione	V
In riferimento allo <i>scheduling</i> di processi, la politiche FIFO ed LRU fanno entrambe parte della categoria degli <i>stack algorithms</i>	F
Un processo per creare un nuovo processo deve fare una <i>system call</i>	V

DOMANDA	Vero/Falso
In un sistema di memoria a paginazione, il <i>Translation Lookaside Buffer</i> (TLB) velocizza la traduzione di indirizzi virtuali in indirizzi fisici	V
La segmentazione consente a due processi di condividere un segmento	V
La paginazione consente a due processi di condividere una pagina	F
Se non vi sono percorsi chiusi in un grafo di allocazione allora non vi è situazione di stallo	V
FAT è un file system ad allocazione concatenata	F
La politica di <i>scheduling round robin</i> minimizza il tempo medio di attesa dei processi	F
Se un processo è in blocco da 10 ms significa che 10 ms fa ha eseguito una <i>system call</i>	V
Ogni <i>interrupt</i> può essere associato ad un processo che ha richiesto una operazione di I/O	F
Con NTFS è possibile che il file system scriva il contenuto di file di piccola dimensione (es. <1KB) direttamente nell'i-node	F
<i>pwd</i> è un comando GNU/Linux per modificare la password	F

DOMANDA	Vero/Falso
In un sistema di memoria a paginazione, il <i>Translation Lookaside Buffer</i> (TLB) velocizza la traduzione di indirizzi fisici in indirizzi virtuali	F
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	V
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	V
Molti <i>page fault</i> su un processo non modificano le prestazioni degli altri processi	F
Il nome e la dimensione sono due attributi di un file	V
NTFS è il file system più utilizzato dai sistemi operativi GNU/Linux	F
Con NTFS è possibile che il file system scriva il contenuto di file di piccola dimensione (es. <1KB) direttamente nel record dell'MFT	V
<i>rmdir</i> è un comando GNU/Linux per rinominare directory	F

DOMANDA	Vero/Falso
Se non vi sono percorsi chiusi in un grafo di allocazione allora non vi è situazione di stallo	V
In un sistema a paginazione la traduzione degli indirizzi mantiene l'offset intatto	V
Il TLB consente di ridurre gli accessi alla tabella delle pagine	V
La segmentazione consente a due processi di condividere un segmento	V
L'ampiezza dello spazio di indirizzamento virtuale dipende dalla dimensione in bit degli indirizzi	V
La strategia di allocazione contigua può comportare frammentazione esterna	V
Per deframmentare un disco occorre prima riformattarlo	F
La politica di <i>scheduling Round Robin</i> non è soggetto al fenomeno di <i>starvation</i>	V
Il meccanismo di <i>context-switch</i> è operato dallo scheduler dei processi	F
Un processo che esegue in spazio kernel non può accedere al suo <i>stack</i> in spazio utente	F

DOMANDA	Vero/Falso
La tecnica di allocazione contigua è soggetta al problema della frammentazione esterna	V
Un indirizzo logico identifica direttamente una cella di memoria fisica	F
Il <i>Process Control Block</i> (PCB) è una struttura dati che memorizza tutte le informazioni riguardanti un processo	V
Le chiamate di sistema comportano il cambiamento della modalità esecuzione da spazio kernel verso spazio utente	F
La tecnica di compattazione della memoria non comporta la rilocalizzazione dei programmi	F
Il principio di località dei riferimenti è la motivazione base che rende efficace l'uso di cache	V
La tecnica di paginazione è soggetta alla frammentazione interna ma non a quella esterna	V
La politica di scheduling <i>Shortest Job First</i> (SJF) non è soggetta al fenomeno di <i>starvation</i>	F
Un <i>page fault</i> avviene quando la entry richiesta non è presente nel <i>Translation Lookaside Buffer</i> (TLB)	F
In un sistema combinato di paginazione/segmentazione lo spazio di indirizzamento utente è suddiviso in un certo numero di pagine a dimensione fissa, che a loro turno sono suddivise in un certo numero di segmenti	F

DOMANDA	Vero/Falso
In un sistema di memoria a paginazione, il <i>Translation Lookaside Buffer</i> (TLB) velocizza la traduzione di indirizzi fisici in indirizzi virtuali	F
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	V
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	V
Molti <i>page fault</i> su un processo non modificano le prestazioni degli altri processi	F
Il nome e la dimensione sono due attributi di un file	V
NTFS è il file system più utilizzato dai sistemi operativi GNU/Linux	F
Con NTFS è possibile che il file system scriva il contenuto di file di piccola dimensione (es. <1KB) direttamente nel record dell'MFT	V
<code>rmdir</code> è un comando GNU/Linux per rinominare directory	F

DOMANDA	Vero/Falso
In un sistema di memoria a paginazione, il <i>Translation Lookaside Buffer</i> (TLB) velocizza la traduzione di indirizzi logici in indirizzi fisici	V
La gestione della memoria con paginazione consente a più processi di condividere pagine contenenti codice o dati	F
<code>chmod</code> è un comando utilizzabile nei sistemi GNU/Linux per modificare i permessi di file e directory	V
GNU/Linux tende a scrivere i file su disco come blocchi sequenziali contigui	V
Con ext2fs è possibile che il file system scriva il contenuto (i dati) di file di piccola dimensione (es. <1KB) direttamente dentro l'i-node principale	F
<code>rmdir</code> è un comando POSIX per eliminare directory vuote	V

Categoria vero/falso A : SECONDA PARTE (GESTIONE MEMORIA).

- 1.** In un sistema di memoria a paginazione, il Translation Lookaside Buffer (TLB) velocizza la traduzione di indirizzi virtuali in indirizzi fisici **V**
- 2.** La segmentazione consente a due processi di condividere un segmento **F**
- 3.** La paginazione consente a due processi di condividere una pagina **F**
- 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 **F**
- 6.** Con NTFS è possibile che il file system scriva il contenuto di file di piccola dimensione (es. <1KB) direttamente nell'inode **V**
- 7.** Pwd è un comando GNU/Linux per modificare la password **F**
- 8.** L'i-node contiene al suo interno i dati dei file (es. il testo scritto da un utente in un file .txt) **F**
- 9.** 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 **F**
- 10.** In un sistema di memoria a paginazione, il Translation Lookaside Buffer (TLB) velocizza la traduzione di indirizzi logici in indirizzi fisici **V**
- 11.** 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 **V**
- 12.** 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 **V**
- 13.** In riferimento allo scheduling di processi, la politica FIFO non causa mai meno page fault della politica LRU **F**
- 14.** Molti page fault su un processo non modificano le prestazioni degli altri processi **F**

Categoria risposte multiple B

- 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
- 2.** 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
- 3.** 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

4. “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)**
5. 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.**
6. 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**
7. 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**
8. 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
9. In un confronto prestazionale tra hard link e symbolic link:
1. **gli HL sono da preferire perché velocizzano gli accessi ai file**
 2. i SL sono da preferire perché assicurano la singolarità dell’associazione tra directory e i-node
 3. i due sono sostanzialmente indistinguibili
 4. i SL hanno prestazioni superiori perché impiegano meno spazio nella directory
10. La dimensione di una FAT dipende da:
1. la quantità di file memorizzati su disco
 2. il numero di partizioni virtuali di un disco
 3. la contiguità con cui sono scritti i file su disco
 - 4. la dimensione 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. Si descriva il funzionamento delle seguenti tre politiche di rimpiazzo: optimal, NRU (Not Recently Used), aging. Possono essere usati anche diagrammi descrittivi.
5. 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 $f(\pi)$ che definisca matematicamente lo spreco di memoria in maniera dipendente dalla dimensione π di una pagina.

– $f(\pi) = (\sigma / \pi) \times \epsilon + \pi / 2$

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

– $\sigma \times \epsilon / \pi^2 + 1/2 = 0 \rightarrow f(\pi) = \sqrt{2 \times \sigma \times \epsilon}$

6. 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
7. La dimensione max di un file ottenibile con ext2fs dipende dalla contiguità con cui sono scritti i blocchi del file su disco?
8. La dimensione max di un file ottenibile con FAT dipende dalla contiguità con cui sono scritti i blocchi del file su disco?
9. La dimensione max di un file ottenibile con NTFS dipende dalla contiguità con cui sono scritti i blocchi del file su disco?
10. 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.

11. 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 risente della contiguità con cui i blocchi di tale file sono disposti sul disco fisso.

RISP: Sì ne risente in quanto l'attributo "dati" di NTFS contiene tante coppie valori quante sono le sequenze contigue su disco. Di conseguenza un file enorme contiguamente scritto su disco richiederà unicamente una coppia valori, mentre un file piccolo composto da N blocchi nessuno contiguo a nessun altro richiederà in MFT un numero N di coppie valori.

- 12.** 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.

RISP: Sì ne risente in quanto gli i-node (principale e di in direzione contenenti puntatori) contengono puntatori a tutti i blocchi del file in questione indipendentemente dalla loro contiguità (è utile che lo studente abbia aggiunto una rappresentazione grafica come quella presente sulle slide o nel libro di testo). Di conseguenza un dato file composto da N blocchi richiederà sempre lo stesso overhead in termini di i-node di indirizzi indipendentemente dalla contiguità del file.

- 13.** 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.

RISP: Molte funzioni del Sistema Operativo sarebbero coinvolte (e stravolte) nel dover gestire un hard-disk di dimensione infinita. Lo studente è invitato a rivisitare criticamente il programma del corso provando a riflettere sulle modifiche necessarie.

- Consideri ad esempio le implicazioni riguardo a: - dimensione degli indirizzi - dimensione strutture dati - utilizzabilità dei file system noti - reperire un file - contiguità file - lista blocchi liberi - importanza o meno della frammentazione - ... (molto altro ancora)

Es: Data la politica RR con quanto q e N processi interattivi tutti con lo stesso comportamento. Ogni interazione da luogo ad un CPU burst che richiede alla CPU per un tempo c . Se: $-c < q$: -un processo aspetta al max $(N-1) \cdot c$ nella coda dei pronti per ottenere la CPU; -un utente aspetta al max $(N-1) \cdot c + c = Nc$ per aspettare che la CPU finisca di elaborare l'interazione; $-c > q$: -considerando che c può essere espresso come $c = a \cdot q + b$ (dove a =quanti di tempo; b =tempo nell'ultimo quanto $(< c)$), un utente aspetta $N \cdot q \cdot a + (N-1) \cdot b$ (oppure $(N+1) \cdot c + a \cdot q$) per iniziare l'ultimo quanto di interazione per l'ultimo processo rimasto ancora attivo.