

Di un sistema è noto che la tabella delle pagine più grande del sistema occupa esattamente 2 frame, il numero di un frame è scritto su 2 byte usando usando tutti i bit a disposizione, e nel sistema sono presenti in media 4 processi che insieme producono una frammentazione interna complessiva media di 4 Kilobyte.

lo spazio logico del sistema è grande: ✗

lo spazio fisico del sistema è grande: ✗

	<input type="text" value="256 Megabyte"/>
<input type="text" value="128 Megabyte"/>	<input type="text" value="4 Megabyte"/>
<input type="text" value="8 Megabyte"/>	<input type="text" value="2 Megabyte"/>
<input type="text" value="nessuno dei valori proposti"/>	<input type="text" value="64 Megabyte"/>

Risposta errata.

La risposta corretta è:

Di un sistema è noto che la tabella delle pagine più grande del sistema occupa esattamente 2 frame, il numero di un frame è scritto su 2 byte usando usando tutti i bit a disposizione, e nel sistema sono presenti in media 4 processi che insieme producono una frammentazione interna complessiva media di 4 Kilobyte.

lo spazio logico del sistema è grande: [4 Megabyte]

lo spazio fisico del sistema è grande: [128 Megabyte]

nel sistema operativo Unix un link simbolico:

Scegli un'alternativa:

- ☐ a. è una sorta di puntatore ad un file (regolare o cartella che sia), ma di tipo particolare, infatti la rimozione del puntatore comporta anche la rimozione del file
- ☐ b. viene implementato allocando un nuovo index node in cui viene scritto il pathname (assoluto o relativo che sia) usato come primo argomento del comando "ln -s".
- ☒ c. è l'unico tipo di link implementabile fra cartelle, dato che se fossero permessi i link fisici si potrebbero creare cicli all'interno del file system ✗
- ☐ d. permette di creare collegamenti veloci e che non occupano spazio aggiuntivo sull'hard disk a file e cartelle di cui si deve fare un uso frequente

Risposta errata.

La risposta corretta è: viene implementato allocando un nuovo index node in cui viene scritto il pathname (assoluto o relativo che sia) usato come primo argomento del comando "ln -s".

Il concetto di "*diritto di prelazione*" può essere applicato sia ad un algoritmo di scheduling che ad un kernel nel suo complesso. Qual è la differenza?

Scegli un'alternativa:

- ☒ a. negli algoritmi di scheduling preemptive un processo utente può essere obbligato dal SO ad abbandonare la CPU. Nei kernel preemptive un processo in kernel mode può essere obbligato ad abbandonare la CPU. ✓
- ☐ b. non c'è nessuna differenza: un algoritmo di scheduling preemptive implica che il sistema operativo su cui è implementato possieda un kernel con diritto di prelazione.
- ☐ c. In un kernel con diritto di prelazione vengono disabilitati gli interrupt quando un processo è in kernel mode, in un algoritmo di scheduling con diritto di prelazione i processi utente possono essere interrotti da altri processi utente
- ☐ d. In un kernel con diritto di prelazione vengono disabilitati gli interrupt quando un processo è in kernel mode, in un algoritmo di scheduling con diritto di prelazione vengono abilitati gli interrupt per permettere il funzionamento del timer hardware

In un sistema operativo che adotta uno scheduling senza diritto di prelazione, quattro processi arrivano al tempo indicato e consumano la quantità di CPU indicata nella tabella sottostante

Processo	T. di arrivo	Burst
Pa	0	8
Pb	2	8
Pc	4	2
Pd	6	1

Se si usa l'algoritmo di scheduling non preemptive che fornisce le migliori prestazioni possibili per schedulare i 4 processi in tabella:

il waiting time medio è: ✓

il turnaround medio è: ✓

il diagramma di GANTT è: ✓

l'algoritmo usato per le risposte precedenti può soffrire di starvation?

✓

Tra le ragioni per cui un sistema operativo può finire in una condizione di thrashing troviamo:

Scegli un'alternativa:

- ☐ a. un elevato numero di processi I/O bound rispetto alla dimensione della memoria principale, e l'adozione di una politica di allocazione globale o locale dei frame
- ☐ b. un elevato numero di processi attivi rispetto alla dimensione della memoria principale, e l'adozione di un algoritmo di sostituzione delle pagine di tipo FIFO
- ☐ c. un elevato numero di processi CPU bound rispetto alla dimensione della memoria principale, e l'adozione di una politica di allocazione globale o locale dei frame
- ☒ d. un elevato numero di processi attivi rispetto alla dimensione della memoria principale, e l'adozione di una politica di allocazione globale o locale dei frame ✓

Secondo quanto visto a lezione, quale/quali dei seguenti comandi Unix modifica il valore del *link counter* dell'index-node associato al file cartella X? (si assuma di avere i permessi per eseguire tutti i comandi e di essere posizionati dentro a X, a sua volta contenuta in una generica cartella user/tmp)

- 1) ln ../X Z
- 2) mkdir K
- 3) ln X Y
- 4) ln -s ../X W

Scegli un'alternativa:

- ☒ a. i comandi 1) e 2) ✖
- ☐ b. il comando 2)
- ☐ c. i comandi 2) 3) e 4)
- ☐ d. i comandi 2) e 4)

Risposta errata.

La risposta corretta è: il comando 2)



Cosa vuol dire che un algoritmo di scheduling soffre di starvation?

Scegli un'alternativa:

- ☐ a. che non garantisce ad un processo di poter uscire da una sezione critica in un tempo finito.
- ☒ b. che non garantisce di assegnare la CPU ad un processo in coda di ready in un tempo finito. ✓
- ☐ c. che non garantisce ad un processo in wait di poter tornare in coda di ready in un tempo finito.
- ☐ d. che non garantisce ad un processo di poter entrare in sezione critica in un tempo finito.

E' noto che il sistema operativo utilizza alcune tecniche per assicurarsi di poter avere sempre sotto controllo il comportamento dei processi utente. Cosa potrebbe succedere se il SO non avesse a disposizione queste tecniche?

Scegli un'alternativa:

- ☐ a. I processi utente potrebbero non riuscire più ad indirizzare le aree di memoria condivisa, ad esempio fondamentali per far funzionare le librerie dinamiche; un processo utente potrebbe non rilasciare mai l'uso della CPU; un processo utente potrebbe usare istruzioni macchina particolarmente delicate, il cui uso errato può danneggiare gli altri processi utente e il sistema operativo.
- ☐ b. non sarebbe più possibile implementare la paginazione della memoria, che è basata su page table gestite esclusivamente dal sistema operativo; un processo utente potrebbe non riuscire mai a disabilitare e riabilitare gli interrupt, necessari per gestire correttamente le sezioni critiche; un processo utente potrebbe usare istruzioni macchina particolarmente delicate, il cui uso errato può danneggiare l'hardware del sistema.
- ☐ c. I processi utente potrebbero indirizzare aree di memoria assegnate ad altri processi, o addirittura quelle riservate al sistema operativo; un processo utente potrebbe venire costretto ad abbandonare l'uso della CPU prima di aver finito il suo lavoro; un processo utente non riuscirebbe più a richiedere l'aiuto del sistema operativo per eseguire operazioni delicate, ad esempio quelle di I/O
- ☒ d. I processi utente potrebbero indirizzare aree di memoria assegnate ad altri processi, o addirittura quelle riservate al sistema operativo; un processo utente potrebbe non rilasciare mai l'uso della CPU; un processo utente potrebbe usare istruzioni macchina particolarmente delicate, il cui uso errato può danneggiare gli altri processi utente e il sistema operativo. ✓

Dopo l'esecuzione dei seguenti comandi in un ambiente Unix (come visti a lezione):

```
1: cd /tmp
2: mkdir newfolder
3: echo "ciao" > pippo // crea un nuovo file di nome pippo contenente la stringa ciao
4: cd newfolder
5: ln ../pippo paperino
6: ln -s /tmp/newfolder folder2
7: cp paperino topolino
8: echo "salve" >> topolino // aggiunge "salve" a fondo file
9: rm pippo
10: cat paperino // cat stampa il contenuto del file passato come argomento
11: mkdir folder3
```

Scegli un'alternativa:

- ☐ a.
 - 1. il link-counter dell'i-node di *paperino* è: 2
 - 2. il link counter di *newfolder* è: 3
 - 3. l'output del comando 10 è: "ciao"
 - 4. il link counter di *tmp* è: aumentato di 1
- ☐ b.
 - 1. il link-counter dell'i-node di *paperino* è: 2
 - 2. il link counter di *newfolder* è: 2
 - 3. l'output del comando 10 è: "ciao"
 - 4. il link counter di *tmp* è: aumentato di 1
- ☒ c.
 - 1. il link-counter dell'i-node di *paperino* è: 2
 - 2. il link counter di *newfolder* è: 3
 - 3. l'output del comando 10 è: no such file or directory
 - 4. il link counter di *tmp* è: aumentato di 1
- ☐ d.
 - 1. il link-counter dell'i-node di *paperino* è: 1
 - 2. il link counter di *newfolder* è: 3
 - 3. l'output del comando 10 è: "ciao"
 - 4. il link counter di *tmp* è: aumentato di 2



Dell'algoritmo di sostituzione delle pagine "Least Recently Used" possiamo dire che:

Scegli un'alternativa:

- ☐ a. fornisce buone prestazioni, vicine a quelle ottimali, sceglie come pagina vittima la pagina entrata in RAM da più tempo, non soffre dell'anomalia di Belady, è difficilmente implementabile e nei processori moderni può essere approssimato usando il reference bit
- ☒ b. fornisce buone prestazioni, vicine a quelle ottimali, sceglie come pagina vittima la pagina in RAM che è stata riferita da più tempo, non soffre dell'anomalia di Belady, è difficilmente implementabile e nei processori moderni può essere approssimato usando il reference bit ✓
- ☐ c. fornisce buone prestazioni, vicine a quelle ottimali, sceglie come pagina vittima la pagina in RAM che è stata riferita da più tempo, non soffre dell'anomalia di Belady, non è implementabile ma nei processori moderni può essere approssimato usando il reference bit
- ☐ d. fornisce buone prestazioni, vicine a quelle ottimali, sceglie come pagina vittima la pagina in RAM che è stata riferita da più tempo, soffre dell'anomalia di Belady, non è implementabile ma nei processori moderni può essere approssimato usando il reference bit