1. Si illustrino le caratteristiche e le problematiche relative alla bufferizzazione in spazio utente, alla bufferizzazione in spazio kernel ed alla doppia bufferizzazione.

   Risposta: In presenza di una semplice bufferizzazione in spazio utente possono sorgere dei problemi con la memoria virtuale; infatti, se la pagina contenente il buffer non si trova in memoria nel momento in cui arrivano i dati dalla periferica di input, il tempo necessario a caricare la pagina dalla memoria secondaria potrebbe essere tale da comportare la perdita di tali dati (che in assenza del buffer non potrebbero essere memorizzati da nessuna parte). Per questo motivo si introduce la bufferizzazione in kernel space; tuttavia ciò comporta la necessità di copiare i dati dallo spazio utente, bloccando nel contempo l'attività di I/O. Inoltre anche in questo caso potrebbero insorgere dei problemi a causa della gestione della memoria virtuale. Infatti ogni qual volta il buffer in spazio kernel si riempie, è necessario copiare i dati nel buffer in spazio utente per renderli utilizzabili al processo e far posto ai nuovi dati in arrivo dalla periferica. Tuttavia, se la pagina in spazio utente contenente il buffer non si trova in memoria, si ripresenta il problema analizzato nel caso precedente, ovvero, la perdita dei dati che continuano ad arrivare dalla periferica mentre la pagina con il buffer viene caricata in memoria principale. La doppia bufferizzazione permette di ovviare a questo problema prevedendo l'allocazione di due buffer in spazio kernel. Quando il primo si riempie, rendendo necessario il trasferimento dei dati in spazio utente, il secondo può essere utilizzato per memorizzare i dati che continuano ad arrivare dalla periferica. Poi i ruoli dei due buffer si invertono e così via.

2. Aumentando il numero di dischi in modo da suddividere il carico di lavoro fra di essi (fornendo all'utente la visione di un disco unitario virtuale), aumenta anche l'affidabilità globale? Perché?

   Risposta: Il semplice fatto di aumentare il numero di dischi in modo da suddividere il carico di lavoro fra di essi (fornendo all'utente la visione di un disco unitario virtuale), non aumenta l'affidabilità globale del sistema in quanto è noto che

   $$MTBF_{array} = \frac{MTBF_{disco}}{\#dischi}$$

   ovvero il tempo medio fra un guasto e l'altro (MTBF=Mean Time Between Failure) diminuisce all'aumentare del numero di dischi in un sistema di calcolo.

3. (a) Cos'è un file?
   (b) Quali sono le operazioni fondamentali sui file che un sistema operativo deve implementare?
   (c) Cosa si intende con l'espressione "file mappati in memoria"? Quali sono i loro vantaggi?

   Risposta:

   (a) Un file è insieme di informazioni correlate a cui è stato assegnato un nome. Esso è inoltre la più piccola porzione unitaria di memoria logica secondaria allocabile dall'utente o dai processi di sistema.

(b) Le operazioni fondamentali sui file che un sistema operativo deve implementare sono le seguenti:

Creazione: allocazione dello spazio sul dispositivo e collegamento di tale spazio al file system.

Cancellazione: staccare il file dal file system e deallocare lo spazio assegnato al file.

Apertura: caricare alcuni metadati dal disco nella memoria pricipale, per velocizzare le chiamate seguenti.

Chiusura: deallocare le strutture allocate nell'apertura.

Lettura: dato un file e un puntatore di posizione, i dati da leggere vengono trasferiti dal media in un buffer in memoria.

Scrittura: dato un file e un puntatore di posizione, i dati da scrivere vengono trasferiti sul media.

Append: versione particolare di scrittura.

Riposizionamento (seek): non comporta operazioni di I/O.

Troncamento: azzerare la lunghezza di un file, mantenendo tutti gli altri attributi.

Lettura dei metadati: leggere le informazioni come nome, timestamp, ecc.

Scrittura dei metadati: modificare informazioni come nome, timestamps, protezione, ecc.

(c) I file mappati in memoria rappresentano un modo efficiente di implementare di implementare le operazioni di lettura/scrittura (soprattutto quando queste sono numerose e coinvolgono uniformemente il contenuto di un file). Praticamente i dati memorizzati nel file vengono copiati (grazie ad una particolare chiamata di sistema) in un'area della memoria principale riservata allo scopo. A questo punto tutte le operazioni di lettura/scrittura su file vengono sostituite da operazioni di lettura/scrittura in memoria: soltanto alla fine di esse il contenuto in memoria verrà salvato su disco. Il vantaggio risultante consiste nell'aumento generale delle prestazioni (vantaggio che risulta tanto più tangibile quanto più numerose sono le operazioni da effettuare su file).

4. Illustrare due soluzioni per evitare l'insorgere di cicli durante la visita di un filesystem con directory a grafo.

   Risposta: Due possibili soluzioni per evitare l'insorgere di cicli durante la visita di un filesystem con directory a grafo sono:

   (a) permettere la creazione di link ai soli file (soluzione adottata per i link hard in UNIX),

   (b) limitare il numero di link attraversabili (soluzione adottata per i link simbolici in UNIX).

5. (a) Si illustri cosa sono l'effective user ID (EUID) e l'effective group ID (EGID) nei sistemi UNIX.

(b) Quali sono le implicazioni sulla sicurezza di un bit `setuid` attivo su un eseguibile posseduto dall'utente `root` e soggetto ad attacchi di tipo buffer overflow?

Risposta:

(a) Nei sistemi UNIX il dominio di protezione di un processo viene ereditato dai suoi figli, e viene impostato al login. In questo modo, tutti i processi di un utente girano con il suo UID e GID. Siccome a volte può essere necessario concedere temporaneamente privilegi speciali ad un utente, sono stati introdotti gli Effective UID e GID (EUID, EGID): due proprietà extra di tutti i processi, in base alle quali vengono determinati i privilegi dei processi di un utente). Normalmente, EUID=UID e EGID=GID, ma possono essere cambiati per la durata dell'esecuzione di un processo attraverso i bit `setuid` e `setgid` dei file eseguibili di UNIX. Se il `setuid` bit è attivo, l'EUID del processo risultante diventa lo stesso del possessore del file. Analogamente, se il `setgid` bit è attivo, l'EGID del processo risultante diventa lo stesso del possessore del file.

(b) Le implicazioni sulla sicurezza di un bit `setuid` attivo su un eseguibile posseduto dall'utente `root` e soggetto ad attacchi di tipo buffer overflow, sono che, se un attaccante riesce a redirigere l'esecuzione su uno shellcode iniettato nello spazio di memoria del processo risultante, si ritroverà a disporre di una shell che gode dei privilegi dell'utente `root`.

6. Si illustri brevemente la struttura di un inode nei sistemi UNIX. Si stimi inoltre la dimensione massima di un file, supponendo di disporre di blocchi da 4 KB e puntatori a 32 bit.

Risposta: Ogni inode in un sistema UNIX contiene i metadati seguenti:

- modo: bit di accesso, di tipo e speciali del le,
- UID e GID del possessore,
- dimensione del le in byte,
- timestamp di ultimo accesso (atime), di ultima modica (mtime), di ultimo cambiamento dellinode (ctime),
- numero di link hard che puntano all'inode,
- blocchi diretti: puntatori ai primi 12 blocchi del le,
- primo indiretto: indirizzo del blocco indice dei primi indiretti,
- secondo indiretto: indirizzo del blocco indice dei secondi indiretti,

In base a ciò segue che la dimensione massima di un file, supponendo di disporre di blocchi da 4 KB e puntatori a 32 bit è

$$\begin{aligned} L_{max} &= 12 + 1024 + 1024^2 + 1024^3 \\ &> 1024^3 = 2^{30} blk \\ &= 2^{42} byte = 4\,TB \end{aligned}$$

dato che con blocchi da 4KB e puntatori a 32 bit(i.e., 4 byte) ogni blocco può contenere 1024 indirizzi.

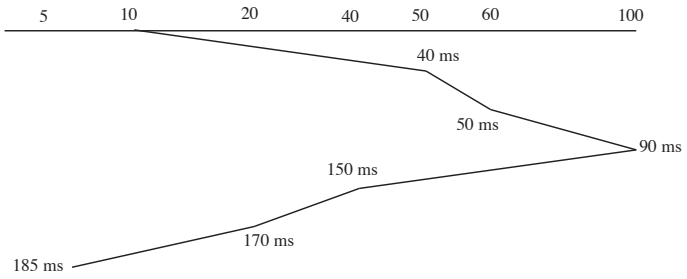
7. Si consideri un disco gestito con politica SCAN. Inizialmente la testina è posizionata sul cilindro 10, ascendente; lo spostamento ad una traccia adiacente richiede 1 ms. Le tracce del disco variano da 0 (prima traccia) a 100 (ultima traccia). Al driver di tale disco arrivano richieste per i cilindri 50, 60, 40, 20, 5, rispettivamente agli istanti 0 ms, 30 ms, 35 ms, 40 ms, 60 ms. Si trascuri il tempo di latenza.

   (a) In quale ordine vengono servite le richieste?

   (b) Il tempo di attesa di una richiesta è il tempo che intercorre dal momento in cui è sottoposta al driver a quando viene effettivamente servita. Qual è il tempo di attesa medio per le quattro richieste in oggetto?

   Risposta:

   (a) Le richieste vengono soddisfatte nell'ordine: 50, 60, 40, 20, 5, come risulta dal seguente diagramma:



   (b) Il tempo di attesa medio per le cinque richieste in oggetto è $\frac{(40-0)+(50-30)+(150-35)+(170-40)+(185-60)}{5} = \frac{40+20+115+130+125}{5} = \frac{430}{5} = 86\ ms$.

Il punteggio attribuito ai quesiti è il seguente: 6, 3, 2, 2, 2, 3, 2, 2, 3, 3, 3 (totale: 31).