- In un sistema operativo multithread, quali risorse sono condivise tra thread dello stesso processo e quali sono private dei singoli thread?

  Risposta: I thread di uno stesso processo condividono le seguenti risorse:

  - spazio di indirizzamento,
  - variabili globali,
  - file aperti,
  - processi figli,
  - timer in scadenza,
  - segnali e routine di gestione dei segnali,
  - informazioni di accounting.

  Le risorse private ad ogni thread invece sono le seguenti:

  - il program counter (PC),
  - i registri,
  - lo stack,
  - lo stato di esecuzione.

- Un algoritmo di scheduling basato su priorità statica è adatto a sistemi time-sharing?

  Risposta: Un algoritmo di scheduling basato su priorità statica non è adatto a sistemi time-sharing in quanto può portare a situazioni in cui l'esecuzione di processi interattivi, ma a bassa priorità sia ritardata eccessivamente (facendo percepire all'utente un tempo di risposta troppo alto e quindi un cattivo servizio) o si arrivi addirittura alla starvation.

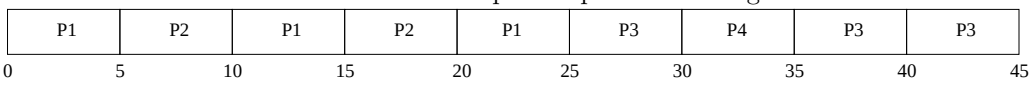
- In coda ready arrivano i processi P_1, P_2, P_3, P_4, con CPU burst e istanti di arrivo specificati in tabella:

	arrivo	burst
P_1	0	15ms
P_2	5	10ms
P_3	15	15ms
P_4	20	5ms

  1. Se i processi terminano nell'ordine P_2, P_1, P_4, P_3, quale può essere l'algoritmo di scheduling? (Si trascuri il tempo di latenza del kernel e, nel caso di processi che arrivano in coda ready nello stesso istante, si dia priorità maggiore al processo con indice inferiore.)
     (a) RR q=10ms
     (b) RR q=5ms
     (c) Scheduling non-preemptive
     (d) SJF
     (e) Nessuno dei precedenti
  2. (*) Si calcoli il tempo di turnaround medio per i processi P_1, P_2, P_3, P_4 della tabella sopra nel caso di algoritmo di scheduling RR q=5ms.

Risposta:

1. a), b).
2. Considerando un algoritmo di scheduling RR q=5ms e trascurando il tempo di latenza del kernel, abbiamo che il diagramma di GANTT relativo all'esecuzione dei quattro processi è il seguente:



e quindi il tempo di turnaround medio è: $\frac{25+15+30+15}{4} = \frac{85}{4} = 21,25ms$.

- Si consideri la seguente situazione, dove P_0, P_1, P_2 sono tre processi in esecuzione, C è la matrice delle risorse correntemente allocate, Max è la matrice del numero massimo di risorse assegnabili ad ogni processo e A è il vettore delle risorse disponibili:

	C			Max		
	A	B	C	A	B	C
P_0	0	2	0	0	5	1
P_1	0	0	0	2	4	3
P_2	2	3	0	2	4	2

Available (A)		
A	B	C
1	5	x

1. Calcolare la matrice R delle richieste.
2. Determinare il minimo valore di x tale che il sistema si trovi in uno stato sicuro.

Risposta:

1. La matrice R delle richieste è data dalla differenza $Max - C$:

R		
A	B	C
0	3	1
2	4	3
0	1	2

2. Se $x = 0$, allora non esiste nessuna riga R_i tale che $R_i \leq A$; quindi il sistema si trova in uno stato di deadlock. Se $x = 1$, allora l'unica riga di R minore o uguale a A è la prima. Quindi possiamo eseguire P_0 che, una volta terminato, restituisce le risorse ad esso allocate aggiornando A al valore $(1, 7, 1)$. A questo punto non esiste alcuna riga di R minore o uguale al vettore A e quindi il sistema è in stato di deadlock. Analogamente, se $x = 2$, dopo aver eseguito P_0, il valore di A è $(1, 7, 2)$. A questo punto si può eseguire P_2 aggiornando A al valore $(3, 10, 2)$: dato che l'unica riga di R rimasta da considerare non è minore o uguale al valore corrente di A, il sistema è in stato di deadlock.

Il valore minimo di x per cui lo stato risulta sicuro è 3; infatti in questo caso esiste la sequenza sicura $\langle P_0, P_2, P_1 \rangle$. Dapprima si esegue P_0, generando il valore $(1, 7, 3)$ di A, poi si esegue P_2 portando A al valore $(3, 10, 3)$. A questo punto si conclude la sequenza eseguendo P_1 e generando il valore finale di A, ovvero, $(3, 10, 3)$.

- Uno dei problemi con una struttura di directory a grafo è la presenza di cicli da evitare durante la visita del filesystem e la creazione di *garbage* in seguito ad operazioni di cancellazione. Si illustrino almeno due possibili soluzioni.

  Risposta: Alcune possibili soluzioni sono le seguenti:

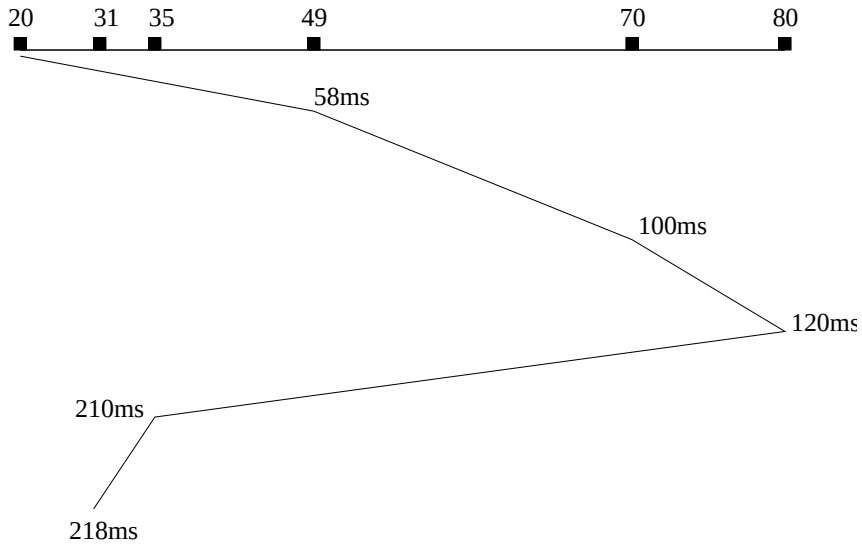
  - permettere la creazione di link ai soli file (soluzione adottata in Unix per i link hard),
  - limitare il numero di link attraversabili (soluzione adottata in Unix per i link simbolici),
  - implementare un meccanismo di garbage collection che marchi durante la visita al filesystem le parti raggiungibili ed elimini tutto ciò che sfugge alla marcatura (bisogna fare attenzione a non visitare più volte gli stessi elementi durante la marcatura per non incappare negli eventuali cicli),
  - verificare la presenza di cicli al momento della creazione di nuovi link (soluzione costosa).

- Spiegare la differenza fra servizi *orientati alla connessione* e servizi *non orientati alla connessione*.

  Risposta: I servizi orientati alla connessione stabiliscono una connessione, ovvero, un canale virtuale per tutta la durata della comunicazione in modo da assicurare il trasferimento dei dati in sequenza. Invece i servizi non orientati alla connessione trasferiscono singoli messaggi senza stabilire e mantenere una vera connessione. In particolare quindi il ricevente può ricevere i messaggi in ordine diverso da quello in cui sono stati spediti.

- (*) Si consideri un disco gestito con politica LOOK. Inizialmente la testina è posizionata sul cilindro 20, ascendente; lo spostamento ad una traccia adiacente richiede 2 ms. Al driver di tale disco arrivano richieste per i cilindri 70, 31, 49, 35, 80, rispettivamente agli istanti 0 ms, 30ms, 40 ms 50 ms, 70 ms. Si trascuri il tempo di latenza. In quale ordine vengono servite le richieste?

  Risposta: L'ordine in cui vengono servite le richieste è 49, 70, 80, 35, 31 come illustrato dal seguente diagramma:



Il punteggio attribuito ai quesiti è il seguente: 3, 3, 3, 3, 2, 2, 4, 4, 6 (totale: 30).