

Cognome e nome: \_\_\_\_\_ Matricola: \_\_\_\_\_ Posto: \_\_\_\_\_

Università degli Studi di Padova – Dipartimento di Matematica. - Corso di Laurea in Informatica

**Regole dell'esame**

Il presente esame scritto deve essere svolto in forma individuale in un tempo massimo di **75 minuti** dalla sua presentazione.

**Non è consentita** la consultazione di libri o appunti in forma cartacea o elettronica, né l'uso di palmari e telefoni cellulari.

Saranno considerati la chiarezza, il rigore dell'esposizione, la capacità di sintesi, la correttezza e completezza delle risposte.

Riportare con chiarezza qualunque ipotesi aggiuntiva ritenuta necessaria alla risoluzione degli esercizi.

La correzione, registrazione ed eventuale sessione orale **avverrà il 20 agosto 2014, ore 14:30, in aula 1BC/50.**

Per superare l'esame il candidato deve acquisire almeno 18 punti su tutti i quesiti, inserendo le proprie risposte interamente su questi fogli. Per la convalida e registrazione del voto finale il docente si riserva di proporre al candidato una prova orale.

**Quesito 0:** Scrivere Cognome e Nome in alto in ogni facciata; scrivere la Matricola e il posto sul primo foglio.

**Quesito 1:**

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"*).

Cognome e nome: \_\_\_\_\_ Matricola: \_\_\_\_\_ Posto: \_\_\_\_\_

**Quesito 2**

Supponiamo di avere 2 processi che condividono una variabile  $x$  e che i loro pseudo-codici siano i seguenti (i numeri a sinistra delle istruzioni non fanno parte del codice, servono solo a identificare le istruzioni nei commenti di chi risolve l'esercizio):

P1:	P2:
1: P (SemA)	12: P (SemB)
2: P (Mutex)	13: P (Mutex)
3: $x=x-2$	14: if ( $x<0$ ) then print ( $x$ )
4: V (Mutex)	15: V (Mutex)
5: V (SemB)	16: V (SemA)
6: P (SemA)	17: P (SemB)
7: P (Mutex)	18: print ( $x$ )
8: $x=x-1$	
9: V (Mutex)	
10: V (SemB)	
11: print ( $x$ )	

I due processi P1 e P2 tentano di eseguire in modo concorrente tra loro. Si assuma che il valore iniziale di  $x$  sia 1 e che i semafori abbiano i seguenti valori iniziali: SemA = 0, SemB = 1, Mutex = 1.

[A] Determinare se e quali istruzioni print ( $x$ ) saranno mai eseguite (indicare il numero alla sinistra dell'istruzione corrispondente) e in caso positivo dichiararne l'output.

[B] Si elenchi inoltre un possibile ordine di esecuzione delle istruzioni appartenenti ai due processi (Es. "1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18").

**Quesito 3**

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).

Cognome e nome: \_\_\_\_\_ Matricola: \_\_\_\_\_ Posto: \_\_\_\_\_

**Quesito 4**

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).

**[4.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.

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

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

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

**[4.E]** Sempre considerando il caso del file di 1MB del punto **4.D**, è possibile che vi sia un i-node che non utilizzerà tutti gli indici a sua disposizione per puntare a dei blocchi. Quanti indici saranno utilizzati e quanti no in questo i-node?

**Cognome e nome:** \_\_\_\_\_ **Matricola:** \_\_\_\_\_ **Posto:** \_\_\_\_\_

**Quesito 5**

Si descriva il funzionamento delle seguenti tre politiche di rimpiazzo: **optimal**, **NRU** (Not Recently Used), **aging**.  
Possono essere usati anche diagrammi descrittivi.

Cognome e nome: \_\_\_\_\_ Matricola: \_\_\_\_\_ Posto: \_\_\_\_\_

**Soluzione****Soluzione al Quesito 1**

Si vedano le slide del corso o il libro di testo.

**Soluzione al Quesito 2**

SOLUZIONE [A]: sarà eseguita solo la print(x) istruzione numero 18 e il risultato stampato sarà -1.

SOLUZIONE [B]: 12, 13, 14, 15, 16, 17, 1, 2, 3, 4, 5, 6, 17, 18

**Soluzione al Quesito 3**

Soluzione desumibile dal materiale spiegato a lezione e durante i laboratori

**Soluzione al Quesito 4**

**[4.A]** La dimensione della partizione di disco non serve a meno di non voler controllare che la dimensione massima del file ottenibili non sfiori questa dimensione; viceversa, l'architettura ext2fs non ne tiene conto in quanto lo spazio occupato dai suoi i-node dipende solo dal numero e dalla dimensione dei file esistenti. Anche l'ipotesi di contiguità del file è inutile in quanto gli i-node contengono gli indici che puntano in sequenza a TUTTI i blocchi di dati di un file a prescindere che questi siano scritti in maniera contigua o meno.

**[4.B]** In questa soluzione useremo la notazione informatica tradizionale, con prefissi che denotano potenze di 2.

Sotto le ipotesi date, la dimensione di un indice è 32 bit = 4 B e il *file* di massima dimensione rappresentabile dall'architettura ext2fs fissata dal quesito sarà composto da:

- 12 blocchi, risultanti dall'utilizzo dei corrispondenti indici diretti presenti nell'i-node principale, al costo di 1 i-node, pari a 1 KB
- $\left\lfloor \frac{1024B}{4B} \right\rfloor = 256$  blocchi, risultanti dall'utilizzo dell'intero i-node secondario denotato dall'indice di I indirezione presente nell'i-node principale, al costo di 1 i-node aggiuntivo, pari a 1 KB
- $256^2 = 2^{16}$  blocchi, risultanti dall'utilizzo dell'indice di II indirezione, al costo di  $1 + 256 = 257$  i-node aggiuntivi, pari a:  $257 \times 1 \text{ KB} = 257 \text{ KB}$

In totale avremo dunque un ammontare di  $12 + 256 + 65536 = 65804$  blocchi di dati ampi 1 KB, corrispondenti a 65804 KB (dimensione massima dei dati in un file)

**[4.C]** Per quanto scritto sopra è evidente che il costo complessivo in termini di spazio occupato dagli i-node è di  $1 + 1 + 257 = 259$  i-node ampi 1 KB

**[4.D]** 1MB di dati corrisponde a 1024 blocchi (di dati). Abbiamo dunque bisogno di 1024 puntatori all'interno della struttura di i-node. Saranno usati i 12 indici diretti presenti nell'i-node principale al costo di tale i-node. Saranno usati i 256 blocchi di prima in direzione al costo di un ulteriore i-node. Rimangono così  $1024 - 12 - 256 = 756$  blocchi per cui predisporre un indice. Usare gli indici di seconda indirezione costa un i-node più quelli necessari puntare ai 756 blocchi, che sono 3: cioè 4.

In totale servono dunque  $1 + 1 + 1 + 3 = 6$  i-node.

**[4.E]** Dei tre i-node di seconda indirezione usati per puntare blocchi di dati, i primi due sono utilizzati completamente per puntare a  $2 \times 256 = 512$  blocchi di dati. Rimangono dunque  $756 - 512 = 244$  blocchi che saranno puntati dal terzo i-node (con dodici indici che rimangono non utilizzati).

**Soluzione al Quesito 5**

Si vedano le slide del corso o il libro di testo.