

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 **90 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.

La correzione, registrazione ed eventuale sessione orale è **fissata al 14 luglio 2014, h.14:30 in aula 1C150**.

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: Illustrare con un diagramma come quello visto a lezione gli stati in cui può trovarsi un processo e le transizioni tra essi. (Presentare solo il diagramma/figura, con i nomi degli stati e delle transizioni; non sono necessarie ulteriori spiegazioni).

Quesito 2

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.

Quesito 3 (Giustificare BREVEMENTE le risposte fornite)

[3.A] La dimensione max di un file ottenibile con ext2fs dipende dalla contiguità con cui sono scritti i blocchi del file su disco?

[3.B] La dimensione max di un file ottenibile con FAT dipende dalla contiguità con cui sono scritti i blocchi del file su disco?

[3.C] La dimensione max di un file ottenibile con NTFS dipende dalla contiguità con cui sono scritti i blocchi del file su disco?

Cognome e nome: _____ Matricola: _____ Posto: _____

Quesito 4

Sia data una partizione di disco ampia **64 GB** organizzata in blocchi dati di ampiezza **1 KB** e indici di dimensione **32 bit**. In caso serva, si consideri l'ipotesi di contiguità **nulla** di un file (ciascun blocco si trova su disco in posizione non adiacente al blocco precedente e a quello successivo nella composizione del file).

[4.A] Si determini l'ampiezza massima di file ottenibile per l'architettura di file system **ext2fs** assumendo 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.

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

[4.C] Si determini la quantità di spazio occupata dal totale della struttura FAT in caso il file system usato per rappresentare il file calcolato in [4.A] nel sistema sopra descritto sia basato su FAT invece che su i-node (ext2fs).

Quesito 5

Si considerino tre processi che alternano fasi di uso di una CPU (*job*) con fasi di I/O *burst*. Si utilizzi una politica di CPU *scheduling* del tipo *shortest remaining time next* (SRTN) applicata sui vari job pronti per l'esecuzione sulla CPU.

I tre processi, denominati P1, P2 e P3, sono caratterizzati dalle seguenti sequenze di CPU *burst* e I/O *burst*.

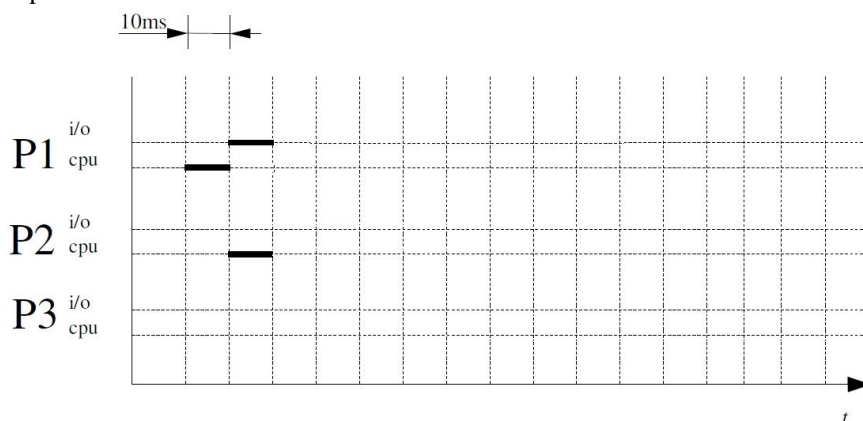
P1: cpu-10ms, i/o-10ms, cpu-30ms, i/o-10ms, cpu-10ms.

P2: cpu-20ms, i/o-10ms, cpu-10ms.

P3: cpu-50ms, i/o-10ms, cpu-10ms.

Ovviamente nessun processo può avanzare alla fase successiva senza aver prima completato le precedenti, nell'ordine.

Si supponga che i tre processi facciano I/O su dispositivi distinti. Si mostri, in ciascun istante di tempo, quali processi risultano in stato di *running* sulla CPU e quali in blocco I/O, marcando, nel diagramma sottostante, le linee di CPU o di I/O corrispondenti a ciascun processo.



Cognome e nome: _____ Matricola: _____ Posto: _____

Quesito 6

Il problema del “produttore/consumatore” è un classico problema di sincronizzazione tra più processi che accedono concorrentemente a risorse condivise. Lo studente utilizzi i **monitor** per scrivere due procedure chiamate `Producer` e `Consumer` che possano essere eseguite concorrentemente al fine di risolvere il problema evitando il *deadlock* del sistema. (Si consideri il caso in cui le risorse prodotte e non ancora consumate possano essere al massimo N).

Quesito 7

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

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

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

Soluzione al Quesito 3

[3.A] No. (Si vedano le slide del corso o il libro di testo per la motivazione).

[3.B] No. (Si vedano le slide del corso o il libro di testo per la motivazione).

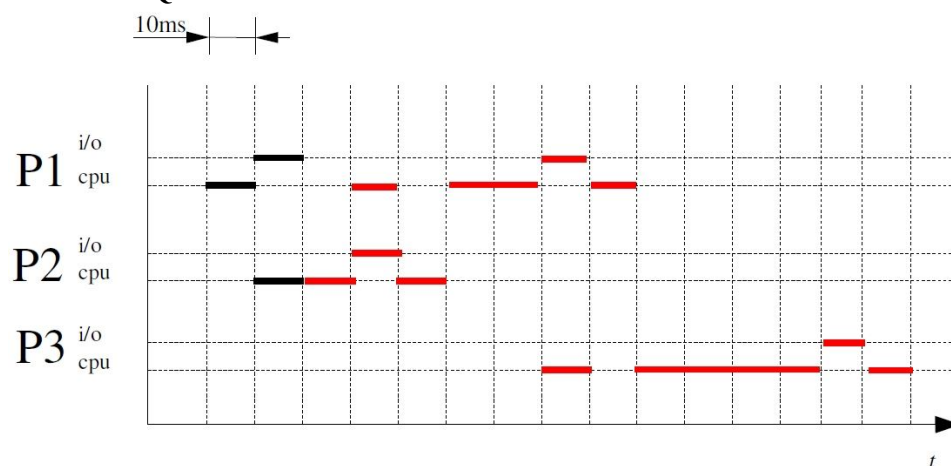
[3.C] Sì. (Si vedano le slide del corso o il libro di testo per la motivazione).

Soluzione al Quesito 4

[4.A] 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
- $\left\lceil \frac{1024B}{4B} \right\rceil = 256$ 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.B] 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.C] La FAT è unica per ogni partizione di disco e non dipende dalla quantità o dimensione dei file in essa contenuti. Essendo la memoria secondaria in questo esercizio ampia 64 GB e i blocchi dati ampi 1 KB, è immediato calcolare che la memoria secondaria è composta di $\left\lceil \frac{64GB}{1KB} \right\rceil = 64 \text{ M} = 2^{26}$ blocchi ognuno dei quali deve avere una entry nella FAT. Siccomeciascuna entry ha dimensione uguale a quella di un indice, ovvero 4 Byte, ne risulta che la dimensione totale della FAT è calcolabile come $= 2^{26} \times 2^2 = 2^{28} = 256 \text{ MB}$.**Soluzione al Quesito 5**

Cognome e nome: _____ Matricola: _____ Posto: _____

Soluzione al Quesito 6

Varie soluzioni possibili, ad esempio:

monitor *ProducerConsumer*

```

condition full, empty;
integer count;
procedure insert(item: integer);
begin
    if count = N then wait(full);
    insert_item(item);
    count := count + 1;
    if count = 1 then signal(empty)
end;
function remove: integer;
begin
    if count = 0 then wait(empty);
    remove = remove_item;
    count := count - 1;
    if count = N - 1 then signal(full)
end;
count := 0;
end monitor;

```

```

procedure producer;
begin
    while true do
    begin
        item = produce_item;
        ProducerConsumer.insert(item)
    end
end;
procedure consumer;
begin
    while true do
    begin
        item = ProducerConsumer.remove;
        consume_item(item)
    end
end;

```

Soluzione al Quesito 7

La soluzione corretta è la 4. Infatti l'Inverted Page Table offre vantaggi rispetto alle Page Table regolari solo in termini di spazio occupato in memoria e non, ad esempio, in termini di performance relative alla velocità di traduzione degli indirizzi da logici a fisici. Il vantaggio maggiore si avrà dunque in corrispondenza della più ampia differenza di occupazione in RAM. Le Page Table hanno un'ampiezza che cresce al crescere della dimensione della Memoria Virtuale, la quale dipende da quanti bit posso usare per esprimere un indirizzo (con N bit posso andare dal byte 0 al byte 2^N).

E' quindi evidente che con 64 bit la page table regolari possono essere molto più grandi che con 32 o meno bit, mentre la dimensione della Inverted Page Table rimane costante in quanto proporzionale alla dimensione della RAM effettiva. Per il caso 64 bit, il vantaggio di usare una Inverted Page Table sarà dunque maggiore che negli altri casi.