

3 aprile 2001

1. (punti: -1,4)

Assumendo in ogni caso un comportamento bloccante per la primitiva **receive**:

- (a) i modelli di comunicazione si dicono sincroni, se la send ha un comportamento bloccante;
- (b) i modelli di comunicazione si dicono sincroni, se la send non ha un comportamento bloccante;
- (c) i modelli di comunicazione si dicono simmetrici, se la send ha un comportamento bloccante;
- (d) i modelli di comunicazione si dicono asimmetrici, se la send non ha un comportamento bloccante;
- (e) la forma di comunicazione chiamata “rendez-vous esteso” richiede un comportamento non bloccante della send;
- (f) la forma di comunicazione chiamata “rendez-vous esteso” richiede un comportamento bloccante della send;
- (g) la forma di comunicazione chiamata “rendez-vous” richiede un comportamento non bloccante della send;
- (h) la forma di comunicazione chiamata “rendez-vous” richiede un comportamento bloccante della send;
- (i) nessuna delle affermazioni precedenti è corretta.

2. (punti: -1,4)

Il meccanismo di traduzione degli indirizzi nella gestione della memoria virtuale con paginazione:

- (a) è basato sull'utilizzo di una memoria associativa, che realizza la ricerca del descrittore di pagina;
- (b) è basato sull'utilizzo di una memoria associativa, che è una memoria veloce che mantiene la tabella di descrittori di pagina del processo in esecuzione;
- (c) le informazioni contenute nella memoria associativa includono: l'indice del blocco in memoria principale nel quale la pagina è caricata;
- (d) le informazioni contenute nella memoria associativa includono: lo stato della pagina caricata;
- (e) le informazioni contenute nella memoria associativa includono: l'indice della pagina caricata;
- (f) utilizza una tabella di descrittori di pagina per ogni processo, i cui elementi sono in corrispondenza con gli accessi in memoria del processo;
- (g) utilizza una tabella di descrittori di pagina unica per tutti i processi, i cui elementi sono in corrispondenza con le pagine dei processi;
- (h) realizza anche la protezione della memoria;
- (i) necessita di meccanismi aggiuntivi per realizzare la protezione della memoria;
- (j) nessuna delle affermazioni precedenti è corretta.

3. (punti: -1,4)

Il sistema di archiviazione di Linux consente di *collegare* (link) un archivio  $F_i$  a più cartelle  $\{D_j, D_h, \dots, D_k\}$ :

- (a) in caso di cancellazione del collegamento simbolico (soft link) da  $D_j$  ad  $F_i$  si richiede una ricerca in tutte le cartelle del sistema di archiviazione per rimuovere da  $\{D_h, \dots, D_k\}$  gli elementi che puntano a  $F_i$ ;
- (b) in caso di cancellazione del collegamento fisico (hard link) da  $D_j$  ad  $F_i$  si richiede una ricerca in tutte le cartelle del sistema di archiviazione per rimuovere da  $\{D_h, \dots, D_k\}$  gli elementi che puntano a  $F_i$ ;
- (c) un collegamento simbolico (soft link) consiste nell'associare il cammino assoluto di  $F_i$  al suo nome in ciascuna delle cartelle  $\{D_j, D_h, \dots, D_k\}$ ;

- (d) un collegamento fisico (hard link) consiste nell'associare il cammino assoluto di  $F_i$  al suo nome in ciascuna delle cartelle  $\{D_j, D_h, \dots, D_k\}$ ;
- (e) un collegamento simbolico (soft link) consiste nel registrare l'*IndicediDescrittore* di  $F_i$  in ciascuna delle cartelle  $\{D_j, D_h, \dots, D_k\}$ ;
- (f) un collegamento fisico (hard link) consiste nel registrare l'*IndicediDescrittore* di  $F_i$  in ciascuna delle cartelle  $\{D_j, D_h, \dots, D_k\}$ ;
- (g) un collegamento simbolico (soft link) consiste nell'associare il cammino relativo di  $F_i$  al suo nome simbolico in ciascuna delle cartelle  $\{D_j, D_h, \dots, D_k\}$ ;
- (h) un collegamento fisico (hard link) consiste nell'associare il cammino relativo di  $F_i$  al suo nome simbolico in ciascuna delle cartelle  $\{D_j, D_h, \dots, D_k\}$ ;
- (i) la *TabellaDeiCollegamenti* contiene, fra l'altro, il numero di collegamenti fisici (hard link) esistenti ad  $F_i$ ;
- (j) la *TabellaDeiCollegamenti* contiene, fra l'altro, il numero di collegamenti simbolici (soft link) esistenti ad  $F_i$ ;
- (k) nessuna delle affermazioni precedenti è corretta.

4. (punti: -1,4)

Quali delle seguenti strutture dati appartengono al sistema di archiviazione di Linux:

- (a) La *TabellaDeiDescrittoriInMS* contiene, fra l'altro, i puntatori alla posizione corrente degli archivi aperti;
- (b) La *TabellaDeiDescrittoriInMS* registra lo stato di assegnazione su disco dei descrittori di archivio;
- (c) La *TabellaDeiDescrittoriInMS* contiene le copie dei descrittori di archivio in uso da parte di almeno un processo;
- (d) La *MappaDeiDescrittoriDiArchivio* registra lo stato di assegnazione su disco dei descrittori di archivio;
- (e) La *MappaDeiDescrittoriDiArchivio* contiene le copie dei descrittori di archivio in uso da parte di almeno un processo;
- (f) La *MappaDeiDescrittoriDiArchivio* contiene, fra l'altro, i puntatori alla posizione corrente degli archivi aperti;
- (g) La *TabellaDeiCollegamenti* contiene le copie dei descrittori di archivio in uso da parte di almeno un processo;
- (h) La *TabellaDeiCollegamenti* registra lo stato di assegnazione su disco dei descrittori di archivio;
- (i) La *TabellaDeiCollegamenti* contiene, fra l'altro, i puntatori alla posizione corrente degli archivi aperti;
- (j) nessuna delle affermazioni precedenti è corretta.

5. (punti: -1,4)

Ad ogni processo del sistema operativo Linux viene associata una quadrupla (*proprietario reale, gruppo reale, proprietario effettivo, gruppo effettivo*):

- (a) il proprietario reale e il gruppo reale contengono informazioni (ad esempio, username) relative all'utente reale, mentre il proprietario effettivo e il gruppo effettivo contengono informazioni (UID e GID) utilizzate internamente dal sistema per identificare in modo univoco l'utente;
- (b) il proprietario reale e il gruppo reale contengono informazioni (UID e GID) utilizzate internamente dal sistema per identificare in modo univoco l'utente, mentre il proprietario effettivo e il gruppo effettivo contengono informazioni (ad esempio, username) relative all'utente reale;

- (c) l'identificazione degli utenti si propaga ai processi, nell'istante di creazione ogni processo è attribuito ad un utente proprietario assegnando opportunamente i valori alla quadrupla che non può mai essere modificata durante l'esecuzione del processo;
- (d) l'identificazione degli utenti si propaga ai processi, nell'istante di creazione ogni processo è attribuito ad un utente proprietario assegnando opportunamente i valori alla quadrupla che può essere modificata durante l'esecuzione del processo;
- (e) i meccanismi di protezione di Linux non consentono la migrazione dei processi in domini di protezione diversi da quelli dei loro proprietari;
- (f) si assuma che un processo invochi la chiamata di sistema **exec** per eseguire il programma contenuto in un archivio  $F_i$ ; **exec** assegna al proprietario reale, e al gruppo reale i valori corrispondenti al proprietario dell'archivio  $F_i$  se e solo se è vera la condizione SETUID contenuta nell' i-node di  $F_i$ ;
- (g) si assuma che un processo invochi la chiamata di sistema **exec** per eseguire il programma contenuto in un archivio  $F_i$ ; **exec** assegna al proprietario effettivo, e al gruppo effettivo i valori corrispondenti al proprietario dell'archivio  $F_i$  se e solo se è vera la condizione SETUID contenuta nell' i-node di  $F_i$ ;
- (h) nessuna delle affermazioni precedenti è corretta.

6. (punti: 6)

Descrivere in al più 80 parole il paradigma di sincronizzazione di processi mediante scambio di messaggi, elencando le varianti conosciute.

7. (punti: 6)

Discutere in al più 70 parole le seguenti transizioni di stato dei processi: sospensione e riattivazione.

8. (punti: 6)

Considerare un insieme di cinque processi  $P_1, P_2, P_3, P_4, P_5$  con i seguenti tempi di arrivo e tempi di esecuzione in millisecondi:

Processo	Tempo di arrivo	Tempo di esecuzione
$P_1$	2	20
$P_2$	17	7
$P_3$	5	8
$P_4$	11	14
$P_5$	22	13

Assegnare questo insieme di processi ad un processore in base alla politica Round Robin considerando un quanto di tempo di 6 millisecondi.

Calcolare il valor medio del tempo di attesa ed il valor medio del tempo di turnaround dei processi.

9. Considerare la seguente stringa di riferimenti alla memoria di un processo in un sistema con memoria virtuale  $S = 1\ 12\ 3\ 2\ 0\ 5\ 2\ 1\ 3\ 9\ 5\ 0\ 9\ 0\ 2\ 10\ 12\ 3\ 9\ 0$

(a) (punti: 3)

Illustrare il comportamento dell'algoritmo LRU di sostituzione delle pagine per una memoria fisica di 5 blocchi. Calcolare il numero di page fault che si verificano.

(b) (punti: 3)

Illustrare il comportamento dell'algoritmo Second Chance di sostituzione delle pagine per una memoria fisica di 5 blocchi. Calcolare il numero di page fault che si verificano.

10. (punti: 16)

Implementare il metodo unlock della seguente classe astratta Java che realizza uno scheduler di risorse condivise. Un orologio di sistema della classe Clk (da assumere come data) fornisce il metodo getCurrentTime() che restituisce l'ora corrente. Lo scheduler risolve eventuali situazioni di stallo invocando il metodo stop() sui processi che rimangono in attesa di un lock per piu' di 100 tic dell'orologio.

```

abstract class Scheduler extends Thread {

    private Clk clock = new Clk();
    private Thread [] Locks;

    /* Locks[i] = T when thread T has the i-th lock */

    public Scheduler (int l) {
Locks = new Thread [l];
for (int i=0; i<l; i++)
    Locks[i] = null;
    }

    public synchronized void lock (int n)
    throws InterruptedException {

int timeout = clock.getCurrentTime() + 100;
while (Locks[n] != null && timeout < clock.getCurrentTime()) {
    wait();
}
if (Locks[n] != null) {
    unlockAll();
    Thread.currentThread().stop();
}
else Locks[n] = Thread.currentThread();
    }

    abstract public void unlock (int n);

    private synchronized void unlockAll () {

for (int i=0; i<Locks.length; i++) {
    if (Locks[i] == Thread.currentThread())
Locks[i] = null;
}
    }

    public void run (){

while (true) {
    try {sleep(20);}
    catch (InterruptedException x) {}
    synchronized(this) {
notifyAll();
    }
}
    }
}

```

**Cognome:**

**Nome:**

**Matricola:**

**Risposte Compito di Sistemi Operativi, 3/4/2001**

1. (punti: -1,4)    **a**    **b**    **c**    **d**    **e**    **f**    **g**    **h**    **i**    **j**    **k**
2. (punti: -1,4)    **a**    **b**    **c**    **d**    **e**    **f**    **g**    **h**    **i**    **j**    **k**
3. (punti: -1,4)    **a**    **b**    **c**    **d**    **e**    **f**    **g**    **h**    **i**    **j**    **k**
4. (punti: -1,4)    **a**    **b**    **c**    **d**    **e**    **f**    **g**    **h**    **i**    **j**    **k**
5. (punti: -1,4)    **a**    **b**    **c**    **d**    **e**    **f**    **g**    **h**    **i**    **j**    **k**
6. (punti: 6)

7. (punti: 6)

8. (a) (punti: 3)

(b) (punti: 3)