

**Università degli studi di Verona**  
**Corso di Laurea in Informatica/Informatica Multimediale**  
**Sistemi Operativi – 29 Marzo 2006**

1. Si consideri il problema del docente impegnato e dello studente preoccupato. Uno studente in vista dell'esame di sistemi operativi desidera incontrare il docente per chiedere delucidazioni sul materiale da studiare. Il docente durante la giornata svolge le seguenti attività: prepara la lezione, mangia, fornisce udienza agli studenti, svolge attività di ricerca. Lo studente può chiedere delucidazioni durante l'orario di ricevimento. Tuttavia, può succedere che il docente non riesca a dare udienza a tutti gli studenti. Se uno studente trova il docente libero durante l'orario di udienza, entra e chiede delucidazioni. Se il docente è impegnato con un altro studente, attende paziente il proprio turno solo se il numero di studenti già in attesa è minore di 3. In tutti gli altri casi (docente non in orario di udienza, o troppi studenti in attesa), lo studente dovrà ripassare un'altra volta. Se durante l'orario di udienza non ci sono studenti, il docente svolge attività di ricerca finché non arriva uno studente. Fornire una soluzione che sincronizzi docente e studenti usando i monitor.

**[8 punti]**

2. Si descriva la struttura del file system nei sistemi Unix e il meccanismo di gestione dei blocchi liberi. Qual'è la dimensione massima indirizzabile se ogni blocco è grande 512KB e gli indirizzi occupano 4 byte?

**[5 punti]**

3. Si descrivano gli algoritmi di scheduling del disco. Mostrare inoltre l'esecuzione di ciascuno con la seguente coda di richieste: 98, 12, 35, 30, 70, 159, 1, 390, 8, 389, 9.

**[5 punti]**

4. Si descriva il concetto di thread evidenziando le differenze tra User-level thread e kernel-level thread.

**[5 punti]**

5. Si supponga di avere una memoria paginata di dimensione 1GB. Se ogni pagina ha dimensione 32KB, quanto è grande l'indirizzo fisico? Quanto è grande l'indirizzo logico? Quanto è grande la tabella delle pagine? Ipotizzando di usare una paginazione multilivello, quanti livelli potremmo avere al massimo per garantire che il tempo effettivo di accesso alla memoria non superi più del 50% il tempo di accesso alla memoria che si avrebbe nel caso di allocazione contigua con partizioni fisse? Motivare le risposte.

**[5 punti]**

6. Si consideri il seguente programma

```
int num;
int vet[10];
void main(int argc, char* argv[]){
    int n = 0;
    while (n < 10){
        scanf("%d", &num);
        vet[n] = f(num);
        printf("vet[%d]=%d\n", n, vet[n]);
        n++;
    }
}
int f(int m){
    int k;
    k = 2*m*m-3;
    return k
}
```

Mostrare in quali parti è divisa la sua immagine in memoria indicando esplicitamente dove vengono inserite le varie istruzioni e dichiarazioni che lo compongono.

**[5 punti]**