

Sistemi Operativi II - Corso di Laurea in Ingegneria Informatica
Appello d'esame del 29/3/2006 - Docente Roberto Baldoni

Cognome _____ Nome _____

Domanda 1 (3pt). Descrivere le tecniche di gestione dei parametri di input e di output nel contesto delle RPC (Remote Procedure Call).

Domanda 2 (3pt). Descrivere lo schema di segmentazione paginata e la relativa traduzione tra indirizzi logici ed indirizzi fisici.

Domanda 3 (3pt). Descrivere gli algoritmi di schedulazione del disco SCAN, CSCAN ed FSCAN discutendone vantaggi e svantaggi. Si consideri inoltre la sequenza di operazioni di I/O corrispondenti alle tracce: 16, 63, 49, 7, 67 e 44. Le operazioni di I/O vengono richieste nell'ordine delle tracce nella sequenza. Supponendo che (1) il disco abbia 100 tracce (numerata da 1 a 100), (2) la testina sia inizialmente posizionata sulla traccia 90, (3) la direzione di movimento della testina per SCAN sia inizialmente decrescente, (4) CSCAN ed FSCAN lavorino con scansione crescente, (4) per FSCAN la sequenza di scheduling corrente comprenda solo le richieste per le tracce 16, 63 e 49, calcolare la lunghezza media di scansione dei tre algoritmi.

Domanda 4 (4pt) Considerare un sistema formato da tre processi concorrenti che competono per entrare nella sezione critica. Determinare se il deadlock è possibile fornendo una sequenza di interleaving che lo produce ed un algoritmo che lo eviti

Mutex M1, M2, M3

P1	P2	P3
while (true) {	while (true) {	while (true) {
1.1 NonCriticalReg();	2.1 NonCriticalReg();	3.1 NonCriticalReg();
1.2 MutexLock(&M1);	2.2 MutexLock(&M2);	3.2 MutexLock(&M3);
1.3 MutexLock(&M2);	2.3 MutexLock(&M3);	3.3 MutexLock(&M1);
1.4 CriticalRegion();	2.4 CriticalRegion();	3.4 CriticalRegion();
1.5 MutexUnlock(&M2);	2.5 MutexUnlock(&M3);	3.5 MutexUnlock(&M1);
1.6 MutexUnlock(&M1);	2.6 MutexUnlock(&M2);	3.6 MutexUnlock(&M3);
1.7 }	2.7 }	3.7 }

Domanda 5- (17 punti)

Il codice che segue è l'implementazione di un programma server che mantiene un contatore numerico condiviso da vari client. Ogni client pu~ richiedere in qualsiasi momento l'incremento di una unità del contatore. Ogni volta che il contatore viene incrementato, il nuovo valore viene stampato a video. Il comportamento atteso è che venga stampata a video sul server una sequenza di numeri crescente senza "salti" o "doppioni". All'uscita dal programma server (tramite la pressione dei tasti Ctrl+C) dovrà essere stampato a video il valore attuale del contatore. Si chiede allo studente di completare il programma e di controllare se questo sia corretto rispetto alla specifica. In caso negativo, discutere eventuali approcci per una

implementazione che soddisfi la specifica.

```
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <signal.h>

int contatore=0;
void incrementa();

int main(int argc, char *argv[]) {

    int servsocket, connsocket, len;
    struct sockaddr_in cliaddr, servaddr;

    printf("Server avviato.\n");
    memset ((char *)&servaddr, 0, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port = htons(3333);

    printf("Inizializzazione socket.\n");

    printf("Armo il segnale.\n");
    signal(SIGUSR1, incrementa);

    while (1) {
        len = sizeof(struct sockaddr_in);

        if (fork()==0) {
            close(servsocket);
            printf("Client connesso.\n");
            kill(getppid(), SIGUSR1);
            close(connsocket);
            exit(0);
        }

        close(connsocket);
    }
}

void incrementa(){
    signal(SIGUSR1, incrementa);
    printf("Incremento il contatore.\n", contatore);
    contatore=contatore+1;
    printf("Il nuovo valore del contatore  %d.\n", contatore);
}
```

Il Sottoscritto, ai sensi della legge 675 del 31/12/96, autorizza il Docente a pubblicare in bacheca e su web i risultati della prova di esame. In fede

Firma leggibile: _____