

## Sistemi Operativi II - Corso di Laurea in Ingegneria Informatica

### Appello d'esame del 22/3/2004

Cognome \_\_\_\_\_ Nome \_\_\_\_\_ RM  FR

*Si raccomanda di scrivere il proprio cognome e nome su questo foglio e di utilizzarlo come cartellina per contenere i fogli con le risposte. Se si considera ambigua una domanda, scrivere la propria interpretazione e rispondere conseguentemente. Chi viene scoperto che copia o consulta appunti verrà espulso dall'aula.*

**AVVERTENZA: rispondere in modo schematico e sintetico, senza essere prolissi. La capacità di sintesi verrà premiata.**

#### **Domanda 1 (4 punti)**

Descrivere i passi principali del processamento da parte del sistema operativo di un pacchetto dati spedito da un utente attraverso un socket ricevuto fino alla sua ricezione da parte della scheda di rete.

#### **Domanda 2 (4 punti)**

Descrivere il funzionamento dell'algoritmo "dell'orologio" per la selezione di vittime nei sistemi a memoria virtuale.

#### **Domanda 3 (5 punti)**

Supponiamo che un SO con tre processi P1, P2, P3 e tre risorse si trovi nel seguente stato S: AVAIL(2,3,4), MAX ((1, 2, 2), (1,2,1), (1,1,1)) e ALLOCATION ((1,2,0), (0,1,1), (1,0,1)) dove le matrici sono descritte per righe. Si determini se S è safe e se la richiesta (0,0,1) fatta dal processo P1 può essere accettata dal sistema operativo.

#### **Domanda 4 (4 punti)**

Supponiamo che un SO gestisce quattro utenti u1, u2, u3, u4 ed un utente root. Gli utenti u1 e u2 appartengono ad un gruppo g1 e u3 ed u4 ad un gruppo g2. Ogni utente è owner di un oggetto. Riempire una access matrix usando l'operazione SWITCH in modo tale che ogni utente di un gruppo può effettuare operazioni di READ/WRITE/EXECUTE sugli oggetti del gruppo e l'utente root possa effettuare operazioni su tutti gli oggetti. Non deve essere possibile effettuare operazioni su oggetti di gruppi diversi e sull'oggetto di root.

#### **Domanda 5 (13 punti)**

Il seguente frammento di codice appartiene ad un server FTP. Un server FTP attende una connessione da parte di un client sulla porta 21 ed interpreta i comandi che il client invia su questo canale. Alla ricezione del comando "PORT X", dove X è un numero intero compreso tra 1024 e 65535, il server apre una nuova connessione TCP verso la porta X del client per instaurare un nuovo canale su cui verranno inviati i dati eventualmente richiesti. Integrare il codice riportato con le linee necessarie per realizzare la funzionalità. Riportare in modo dettagliato in quale punto del codice le linee aggiuntive devono essere inserite.

```

4 int list_s, data_s, control_conn;
5 int main(int argc, char *argv[]) {
6     short int          listen_port;
7     struct sockaddr_in servaddr;
8     struct sockaddr_in their_addr;
9     struct sockaddr_in client_addr;
10    char                buffer[MAX_LINE], client_port[6];
11    char                *endptr;
12    int                 sin_size;
13
14    listen_port = 21;
15    printf("Server in ascolto sulla porta %d.\n",listen_port);
16
17    /* Creazione del socket */
18    printf("server: creazione socket.\n");
19    if ( (list_s = socket(AF_INET, SOCK_STREAM, 0)) < 0 ) {
20        fprintf(stderr, "server: errore creazione socket.\n");
21        exit(EXIT_FAILURE);
22    }
23
24    /* Inizializzazione struttura sockaddr_in */
25    printf("server: init struttura sockaddr.\n");
26    memset(&servaddr, 0, sizeof(servaddr));
27    servaddr.sin_family = AF_INET;
28    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
29    servaddr.sin_port = htons(listen_port);
30
31    /* Bind del socket sull'indirizzo IP */
32    printf("server: bind del socket.\n");
33    if (bind(list_s,(struct sockaddr *) &servaddr, sizeof(servaddr))<0) {
34        fprintf(stderr, "server: errore durante la bind.\n");
35        exit(EXIT_FAILURE);
36    }
37
38    /* Poniamo il socket in modalita' passiva */
39    printf("server: listen socket.\n");
40    if ( listen(list_s, LISTENQ) < 0 ) {
41        fprintf(stderr, "server: errore durante la listen.\n");
42        exit(EXIT_FAILURE);
43    }
44
45
46
47    /* Loop infinito */
48    while ( 1 ) {
49        /* da completare */
50    }
51 }

```

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: \_\_\_\_\_