

Sistemi Operativi

Soluzione della Prova Scritta del 6 Febbraio 2007

Esercizio 1 (6 punti)

Illustrate il modello a stati usato per rappresentarne l'esecuzione di un processo da parte del sistema operativo. In particolare, e illustrate chiaramente il significato di ciascuno stato (3 punti) e degli eventi che causano le transizioni fra gli stati.

Esercizio 2 (6 punti)

Mostrare la sequenza di esecuzione dei job riportati in tabella nel caso si usino rispettivamente uno scheduling *Shortest Remaining Time (SRT)* (2 punti) o *round-robin* con feedback con 2 code con quanti di tempo pari rispettivamente a 3 e 6 (2 punti). Calcolare i tempi di turnaround medio e di turnaround normalizzato medio e commentare il risultato ottenuto (2 punti).

Processo	Tempo di Arrivo	Tempo di Elaborazione
A	0	6
B	2	3
C	6	2
D	8	6

SRT	A	A	B	B	B	A	C	C	A	A	A	D	D	D	D	D	D
RR Feedback	A	A	A	B	B	B	C	C	D	D	D	A	A	A	D	D	D

		A	B	C	D	
	T_a	0	2	6	8	
	T_s	6	3	2	6	
SRT	T_f	11.00	5.00	8.00	17.00	
	T_r	11.00	3.00	2.00	9.00	6.25
	T_r/T_s	1.83	1.00	1.00	1.50	1.33
RR-Feedback	T_f	14.00	6.00	8.00	17.00	
	T_r	14.00	4.00	2.00	9.00	7.25
	T_r/T_s	2.33	1.33	1.00	1.50	1.54

Esercizio 3 (5 punti)

Illustrare la funzione realizzata dal seguente programma, commentandone il codice.

```
#include <iostream>
#include <cstdio>
#include <unistd.h>
#include <string.h>
using namespace std;
int main(int argc, char *argv[ ]) {
    int          f_des_par[2];
    int          f_des_child[2];
    static char  message[BUFSIZ];
    char        reply[]="Messaggio ricevuto";

    pipe(f_des_p);
    pipe(f_des_c);

    switch (fork( )) {
        case 0:
            close(f_des_p[1]);
            close(f_des_c[0]);
            read(f_des_p[0], message, BUFSIZ);
            cout << "The message is: [" << message << "]" << endl;

            write(f_des_c[1], reply, strlen(reply));
            break;

        default:
            close(f_des_p[0]);
            close(f_des_c[1]);
            write(f_des_p[1], argv[1], strlen(argv[1]));

            read(f_des_c[0], message, BUFSIZ);
            cout << "The message is: [" << message << "]" << endl;
    }
    return;
}
```

Esercizio 4 (7 punti)

Si consideri il problema dei “Filosofi a tavola”, di cui sotto si riporta una possibile soluzione basata sull’uso del costrutto “monitor”

1. Illustrare la formulazione del problema dei filosofi a tavola, evidenziando le problematiche di mutua esclusione, sincronizzazione e le possibili cause di stallo e starvation (3 punti)
2. Commentare la soluzione al problema riportata sotto evidenziando il grado di multiprogrammazione, e la possibilità che si verifichino condizioni di stallo o starvation (4 punti)

```
monitor dining_controller;
enum states {thinking, hungry, eating} state[5];
cond needFork[5]

void get_forks(int pid) /* pid is the philosopher id number */
{
    state[pid] = hungry;
    if (state[(pid+1) % 5] == eating || (state[(pid-1) % 5] == eating))
        cwait(needFork[pid]);
    state[pid] = eating;
}

void release_forks(int pid)
{
    state[pid] = thinking;
    if (state[(pid+1) % 5] == hungry) || (state[(pid+2) % 5] != eating)
        csignal(needFork[pid+1]);
    else if (state[(pid-1) % 5] == hungry) || (state[(pid-2) % 5] != eating)
        csignal(needFork[pid-1]);
}
```

```
void philosopher[k=0 to 4] /* the five philosopher clients */
{
    while (true)
    {
        <think>;
        get_forks(k);
        <eat spaghetti>;
        release_forks(k);
    }
}
```

Soluzione

Vedi libro di testo

Esercizio 5 (5 punti)

Si consideri la seguente lista di richieste di accessi ad un disco in termini di n.ro di tracce: 27, 129, 110, 186, 147, 41, 10, 64, 120. Si ipotizzi che inizialmente la testina si trovi posizionata sulla traccia 100 e si stia muovendo in direzione di n.ri di traccia decrescenti. Mostrare in quale sequenza vengono servite le richieste usando la politica C-SCAN con tecnica LOOK. Dire inoltre quale è il numero medio di tracce che vengono attraversate.

C-SCAN	
Sequenza di attraversamento delle tracce	Numero di tracce attraversate
64	36
41	23
27	14
10	17
186	176
147	39
129	18
120	9
110	10
Media	38

Esercizio 6 (4 punti)

Illustrare in modo chiaro e sintetico le caratteristiche che deve possedere un sistema operativo per poter essere classificato come "fidato" (*trusted*)