

Esame di Sistemi Operativi

AA 2020/21

04 Giugno 2021

Nome	Cognome	Matricola

Esercizio 1

Scrivere un breve programma per arduino che reagisca all'interruzione INT0 (pin 0 della porta D), e commuti il valore del led (pin 7 della porta B)

```
1  ISR(INT0_vect) {
2      //toggle status of bit 7 of port b
3      // FILL HERE
4  }
5
6  void main() {
7      GICR=0x40;    //Enable External Interrupts INT0
8      MCUCR=0x00;   //Configure INT0 active low
9      // configure ports (D as input, pull up, B as output on pin 7)
10     // FILL HERE
11     sei();
12     while(1) {
13         sleep_cpu();
14     }
15 }
16
```

```
1  uint8_t mask=0xC0;
2  ISR(INT0_vect) {
3      //toggle status of bit 7 of port b
4      if (PORTB&mask){
5          PORTB|= (~mask);
6      } else {
7          PORTB&= mask;
8      }
9  }
10
11 void main() {
12     GICR=0x40;    //Enable External Interrupts INT0
13     MCUCR=0x00;   //Configure INT0 active low
14     // configure ports (D as input, pull up, B as output on pin 7)
15     // FILL HERE
16     DDRB&= (~mask);
17     PORTB= mask;
18     sei();
19     while(1) {
20         sleep_cpu();
21     }
22 }
23
```

Nome	Cognome	Matricola

Esercizio 2

Si consideri in sottosistema di memoria il caratterizzato dalla tabella delle pagine

Page	Frame
0x00	0x10
0x01	0x0F
0x02	0x0E
0x03	0x0D
0x04	0x0C
0x05	0x0B
0x06	0x0A
0x07	0x09
0x08	0x08
0x09	0x07
0x0A	0x06
0x0B	0x05
0x0C	0x04
0x0D	0x03
0x0E	0x02
0x0F	0x01
0x10	0x00
0x11-0xFE	invalid
0xFF	0x10

Domanda Assumendo che le i frame abbiano una dimensione di 4 Kbyte, che la tabella delle pagine consista di 256 elementi, come vengono tradotti in indirizzi fisici i seguenti indirizzi logici?

- 0x00CA1
- 0x04FAC
- 0x011FF
- 0xFFCA1

Soluzione

- 0x00CA1 - 0x10CA1
- 0x04FAC - 0x0CFAC
- 0x011FF - 0x0F1FF
- 0xFFCA1 - 0x101FF

Nome	Cognome	Matricola

Esercizio 4

Illustrare nel dettaglio come puo' essere implementato il meccanismo di context switch.

Soluzione Vedi slides 18-27 slides “syscalls-dual-mode.odp”. Basta commentare il postambolo (slide 27).

Nome	Cognome	Matricola

Esercizio 5

Illustrare l'algoritmo "second chance" per la scelta dei frame da rimuovere in un sistema a memoria virtuale.

Soluzione E' necessario un supporto hardware nella MMU che setti ad 1 un bit nella entry corrispondente ad ogni pagina acceduta. E' necessario un indice nella tabella delle pagine, che tenga traccia della pagina successiva a quella che ha subito per ultima uno swap out. Quando necessario parte una scansione della tabella delle pagine, a partire dall'indice corrente, incrementandolo. Se la pagina oggetto della verifica e' stata acceduta, si setta il bit di accesso a 0, e si incrementa l'indice. Se la pagina non e' stata acceduta sara' la vittima selezionata, si incrementa l'indice al successore e l'algoritmo termina. Quando l'indice raggiunge il limite della tabella delle pagine viene resettato a 0. Presenta l'anomalia di Belady.

Nome	Cognome	Matricola

Esercizio 6

Sia dato il seguente programma:

```

1  #define STACK_SIZE 16384
2  #define ITERATIONS 5
3  ucontext_t main_context, f1_context, f2_context;
4
5  void f1(){
6      for (int i=0; i<ITERATIONS; i++) {
7          printf("f1: %d\n", i);
8          swapcontext(&f1_context, &f2_context);
9      }
10     setcontext(&main_context);
11 }
12
13 void f2(){
14     for (int i=0; i<ITERATIONS; i++) {
15         printf("f2: %d\n", i);
16         swapcontext(&main_context, &f2_context);
17     }
18     swapcontext(&f2_context, &f1_context);
19 }
20
21 char f1_stack[STACK_SIZE];
22 char f2_stack[STACK_SIZE];
23
24 int main() {
25     getcontext(&f1_context);
26
27     f1_context.uc_stack.ss_sp=f1_stack;
28     f1_context.uc_stack.ss_size = STACK_SIZE;
29     f1_context.uc_stack.ss_flags = 0;
30     f1_context.uc_link=&main_context;
31     makecontext(&f1_context, f1, 0, 0);
32
33     f2_context=f1_context;
34     f2_context.uc_stack.ss_sp=f2_stack;
35     f2_context.uc_stack.ss_size = STACK_SIZE;
36     f2_context.uc_stack.ss_flags = 0;
37     f2_context.uc_link=&main_context;
38     makecontext(&f2_context, f2, 0, 0);
39
40     swapcontext(&main_context, &f1_context);
41     printf("exiting\n");
42 }
43

```

Domanda Cosa stampa il programma?

Soluzione

F1
F2
F2
..
F2

Il programma non termina. Il primo cambio di contesto e' main-f1. Il secondo e' f1-f2. In f2, la variabile dove il contesto e' salvata e' main_context, e si salta di nuovo al contesto f2 corrispondente all'iterazione 0. Pertanto parte un ciclo infinito.

Nome	Cognome	Matricola

Esercizio 7

In cosa consiste il fenomeno del trashing? Che relazione c'è tra il working set di un processo e l'insorgenza del "trashing"?

Soluzione Slides 20-21-22 del blocco "virtual-memory".

Nome	Cognome	Matricola

Esercizio 8

Cosa succede quando si invoca una system call? Illustrare tutti i passi, dalla compilazione dei registri della CPU all'esecuzione della routine che implementa la syscall.

Soluzione

- si riempiono i registri della cpu secondo la calling convention del sistema operativo.
- si esegue una trap alla call di sistema (INT 80 su X86/linux)
- in kernel mode, il gestore delle syscalls salverà lo stato del processo, ed individuerà attraverso il numero di syscall la routine opportuna memorizzata in un vettore di puntatori a funzione (o strutture che li contengono). Il numero di syscall corrisponde alla posizione nel vettore della routine. Alla sua terminazione, il gestore delle syscalls terminerà configurando la CPU per continuare l'esecuzione del processo (quello che ha invocato la syscall od un'altro).