

Sistemi di Calcolo (A.A. 2025-2026)

Corso di Laurea in Ingegneria Informatica e Automatica
Sapienza Università di Roma

B

Compito (22/07/2026) – Durata 1h 30'

Inserire nome, cognome e matricola nel file `studente.txt`.

ISTRUZIONI PER STUDENTI DSA: svolgere a scelta due parti su tre.

Parte 1 (programmazione IA32)

Nel campo della security è fondamentale poter filtrare rapidamente gli eventi in base alla loro criticità. A tale scopo, è utile disporre di un meccanismo in grado di assegnare uno score di rischio a ciascun evento e di calcolare un indicatore complessivo del rischio associato a una sequenza di eventi. In questo esercizio, si chiede di tradurre in assembly IA-32 una funzione che, dato un insieme `events` di eventi identificati da valori `short`, la sua dimensione `n`, due soglie `low_th` e `high_th` e un puntatore `total`, calcola il fattore di rischio totale. Il risultato viene scritto nella variabile puntata da `total`. Ogni evento contribuisce al totale in base allo score restituito dalla funzione `risk_score` e al superamento delle soglie indicate. Nella directory E1, si traduca in assembly IA32 la seguente funzione C scrivendo un modulo `e1A.s`:

```
void suspicious_events_score(short* events, int n, int low_th, int
high_th, int* total)
{
    int i;
    int score;
    *total = 0;

    for (i = 0; i < n; i++) {
        score = risk_score((int)events[i]);
        if (score > low_th) {
            *total += 1;
        }
        if (score > high_th) {
            *total += 1;
        }
    }
}
```

L'unico criterio di valutazione è la correttezza. Generare un file eseguibile `e1B` con `gcc -m32 -g`. Per i test, compilare il programma insieme al programma di prova `e1B_main.c`.

Non modificare in alcun modo `e1B_main.c`. Prima di tradurre il programma in IA32 si suggerisce di scrivere nel file `e1B_eq.c` una versione C equivalente più vicina all'assembly.

Parte 2 (programmazione di sistema POSIX)

Si scriva una funzione multi-processo che verifichi la simmetria di un array rispetto al suo centro (array palindromo). Ad esempio:

[1, 2, 3, 4, 3, 4, 1] è simmetrico [1, 2, 3, 4] non è simmetrico in posizione 0

Nello specifico, si scriva nel file `E2/e2B.c` una funzione con il seguente prototipo:

```
int check_symmetry(const int * input, const unsigned short length)
```

che, dato un array di interi `input` di lunghezza `length` restituisce `-length-1` se l'array è simmetrico rispetto al suo centro, oppure l'indice della prima posizione per cui non è valida la proprietà di simmetria. Un array vuoto è per definizione simmetrico.

La soluzione deve fare uso di più processi, ognuno dei quali deve confrontare una posizione di indice `i` con la rispettiva posizione di indice `length-i-1`. **Attenzione:** si suggerisce di trasmettere il risultato di un processo figlio al padre tramite la funzione `exit(status)` dove `status` sarà posto uguale al risultato del confronto. Nel caso in cui l'array non fosse simmetrico la ricerca della posizione può essere calcolata successivamente nel processo padre.

Per i test, compilare il programma insieme al programma di prova `e2B_main.c` fornito, che **non** deve essere modificato.

Parte 3 (quiz)

Si risponda ai seguenti quiz, inserendo le risposte (A, B, C, D o E per ogni domanda) nel file `e3A.txt`. Una sola risposta è quella giusta. Rispondere E equivale a non rispondere (0 punti).

Domanda 1 (Memoria virtuale)

Si consideri un sistema di memoria virtuale con spazio di indirizzi (logico e fisico) a 24 bit e pagine da 256 KB. Data la seguente tabella delle pagine (indice pagina → frame, valori esadecimali), a quali indirizzi fisici corrispondono gli indirizzi logici `0x15C3F0`, `0xAAA0B4`, `0xFCFFF0`?

idx	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0_	07	0C	11	16	1B	20	25	2A	2F	34	39	3E	03	08	0D	12
0x1_	17	1C	21	26	2B	30	35	3A	3F	04	09	0E	13	18	1D	22
0x2_	27	2C	31	36	3B	00	05	0A	0F	14	19	1E	23	28	2D	32
0x3_	37	3C	01	06	0B	10	15	1A	1F	24	29	2E	33	38	3D	02

A	<code>0x51C3F0</code> , <code>0x66A0B4</code> , <code>0x08FFF0</code>	B	<code>0x81C3F0</code> , <code>0x66A0B4</code> , <code>0x08FFF0</code>
C	<code>0x81C3F0</code> , <code>0x6AA0B4</code> , <code>0x08FFF0</code>	D	<code>0x81C3F0</code> , <code>0x66A0B4</code> , <code>0x88FFF0</code>

Motivare la risposta nel file `M1.txt`. **Risposte non motivate saranno considerate nulle.**

Domanda 2 (File system e permessi)

Un utente invoca sul terminale Linux il comando `chmod 750 documento.txt`. Quali diritti di accesso effettivi sono stati assegnati rispettivamente al Proprietario (Owner), al Gruppo (Group) e agli Altri utenti (Others)?

A	Owner: lettura e scrittura; Group: sola lettura; Others: nessun permesso	B	Owner: lettura, scrittura ed esecuzione; Group: lettura ed esecuzione; Others: nessun permesso
C	Owner: lettura, scrittura ed esecuzione; Group: lettura e scrittura; Others: sola lettura	D	Owner: sola esecuzione; Group: lettura e scrittura; Others: nessun permesso

Motivare la risposta nel file `M2.txt`. **Risposte non motivate saranno considerate nulle.**

Domanda 3 (Segnali)

Quale di questi segnali non può essere gestito con una `sigaction`?

A	SIGALRM	B	SIGCHLD
C	SIGKILL	D	SIGSEGV

Motivare la risposta nel file `M3.txt`. **Risposte non motivate saranno considerate nulle.**

Domanda 4 (Allocazione di memoria)

Si consideri un semplice allocatore di memoria dove, potendo scegliere, la `malloc` riusa il blocco libero con l'indirizzo più basso. Assumere per semplicità che non vi sia alcun header e che i blocchi allocati vengono arrotondati a una dimensione multiplo di 4 byte. Inoltre, l'allocatore non divide blocchi liberi in più blocchi di dimensione inferiore, né fonde eventuali blocchi liberi adiacenti in un blocco di dimensioni superiori. Qual è il contenuto dell'heap alla fine della seguente sequenza di operazioni ("X" denota 4 byte allocati e "." denota 4 byte liberi)?:

```
p1 = malloc(4)
p2 = malloc(6)
p3 = malloc(11)
p4 = malloc(3)
free(p2)
free(p4)
p5 = malloc(5)
p6 = malloc(13)
```

A	X . XXX . XX XXXX	B	X XX XXX . XXXX
C	X XX XXX . XXX	D	Nessuna delle precedenti

Motivare la risposta nel file `M4.txt`. **Risposte non motivate saranno considerate nulle.**