

Sistemi di Calcolo (A.A. 2024-2025)

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



Compito (19/06/2025) – Durata 1h 30'

Inserire nome, cognome e matricola nel file `studente.txt`. Se il file non viene correttamente compilato, il compito non sarà valutato.

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

Parte 1 (programmazione IA32)

La funzione `fight` simula una battaglia tra Pokemon. Ogni Pokemon ha un certo numero di punti ferita `pf`, un attacco base ed uno più potente che attiva ogni `countdown` cicli. Queste informazioni sono raccolte in una struttura definita nel file `e1A.h`. Il combattimento avviene in non più di 10 round e un Pokemon vince quando i punti ferita `pf` dell'avversario scendono a 0 o diventano negativi. Nel caso in cui i punti ferita di entrambi gli oppositori diventino negativi contemporaneamente, oppure siano passati i 10 round senza che un Pokemon sia sconfitto, il combattimento si risolve in un pareggio. La funzione `fight` fa uso di una funzione `update` per aggiornare i `pf` di un Pokemon nel turno `i` quando attaccato dall'altro Pokemon e di una funzione `winner` che controlla se un Pokemon ha vinto nel turno, restituendo rispettivamente i valori 1 o 2, entrambi hanno esaurito i punti ferita, restituendo il valore 0, oppure la battaglia deve continuare, restituendo il valore -1.

In questo esercizio, si chiede di tradurre in assembly la funzione `fight` scrivendo il relativo codice nel file `e1A.s` presente nella directory `E1`.

```
unsigned char fight(pokemon* p1, pokemon* p2){
    short pf1 = p1->pf;
    short pf2 = p2->pf;
    char res = 0;

    unsigned char i;
    for (i = 1; i <= 10; i++) {
        pf2 = update(i, p1, pf2);
        pf1 = update(i, p2, pf1);
        res = winner(pf1, pf2);
        if (res >= 0)
            break;
        res = 0;
    }
    return (unsigned char) res;
}
```

L'unico criterio di valutazione è la correttezza. Generare un file eseguibile `e1A` con `gcc -m32 -g`. Per i test, compilare il programma insieme al programma di prova `e1A_main.c`.

Non modificare in alcun modo `e1A_main.c` e `e1A.h`. Prima di tradurre il programma in IA32 si suggerisce di scrivere nel file `e1A_eq.c` una versione C equivalente più vicina all'assembly. Il contenuto del file `e1A_eq.c` NON sarà oggetto di valutazione.

Parte 2 (programmazione di sistema POSIX)

Si scriva una funzione che processi parallelamente un file di testo suddiviso in paragrafi. Nello specifico, si scriva nel file `E2/e2A.c` una funzione con il seguente prototipo (definito in `E2/e2A.h`):

```
void numeroFraseMaxParagrafi(const char* filename, int*
maxNumeroFrase);
```

che, dato in input il nome `filename` del file contenente del testo, ritorni nella variabile puntata da `maxNumeroFrase`, un valore intero contenente il massimo numero di frasi contenuti in uno dei paragrafi presenti nel file. Il conteggio delle frasi presenti in ogni paragrafo dovrà avvenire **in modo parallelo**, ossia in modo tale che ogni paragrafo venga elaborato da un **processo figlio differente**. Paragrafi distinti sono separati da una linea vuota. Ciascuna frase in un paragrafo termina con il carattere `'.'`.

Per esempio, se `filename` contenesse:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Avremmo 2 paragrafi che dovranno essere analizzati in 2 processi distinti. Il primo processo conterà le 2 frasi presenti nel primo paragrafo, mentre il secondo processo conterà invece una sola frase presente nel secondo paragrafo. La funzione, dopo aver atteso il completamento dei processi generati, scriverà quindi il valore 2 nella variabile puntata da `maxNumeroFrase`.

Si ricordi che il carattere di ritorno a capo è rappresentato da `'\n'`. Si assuma che il numero massimo di paragrafi in `filename` sia 1024, che la lunghezza massima di un paragrafo sia 10000 byte e che una riga non sia più lunga di 1024 byte (si possono utilizzare le macro definite in `e2A.h`).

Per i test, compilare il programma insieme al programma di prova `e2A_main.c` fornito, che **non** deve essere modificato. Inoltre, **non modificare per nessun motivo** i file `lipsum*.txt` che riportano esempi di testo.

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 (assembly)

Se `%ax` vale `0x0010`, quale valore assume dopo l'esecuzione di `shlw $2, %ax`?

A	<code>0x0014</code>	B	<code>0x0030</code>
C	<code>0x0040</code>	D	<code>0x1000</code>

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

Domanda 2 (memoria)

Considerata una struct con quattro campi, quale tra i seguenti layout occupa meno memoria?

A	char, short, char, int	B	short, char, int, char
C	char, char, int, short	D	int, short, char, char

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

Domanda 3 (paginazione)

Un sistema ha spazio di indirizzamento virtuale a 28 bit e una dimensione di pagina di 8 KB. Sapendo che ogni entry della tabella delle pagine è lunga 4 byte, quanto spazio occupa la tabella delle pagine per un singolo processo?

A	1 MB	B	512 KB
C	128 KB	D	Nessuna delle precedenti

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

Domanda 4 (pipeline)

Si consideri la seguente sequenza di istruzioni che vengono eseguite da una CPU con pipeline a 5 stadi (Fetch, Decode, Execute, Memory, Write-Back)

```
subl $8, %esp
movl %esp, %eax
addl $4, %eax
movl $0x41414141, %(esp)
movl $0x42424242, %(eax)
```

Quanti cicli di clock vengono richiesti per completare tutte le istruzioni assumendo che gli hazard vengano risolti con stalli?

A	18	B	10
C	11	D	20

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