Politecnico di Milano – Sede di Cremona
Anno Accademico 2020/2021

Architettura dei Calcolatori e Sistemi Operativi

Esame – 18.02.2022

Prof. Carlo Brandolese

Cognome _____ **Nome** _____

Matricola _____ **Firma** _____

Istruzioni

1. Scrivere con cura, negli spazi sopra segnati, il proprio cognome, nome, numero di matricola e apporre la firma.
2. È vietato consultare libri, eserciziari, appunti ed utilizzare la calcolatrice e qualunque strumento elettronico (inclusi i cellulari), pena l'invalidazione del compito.
3. Il testo, debitamente compilato, deve essere riconsegnato in ogni caso.
4. Il tempo della prova è di 3 ore

Valutazione

Domanda	Voto	Note
A		
B		
C		
D		
E		
F		

Domanda A

Si implementi in codice assembly la funzione:

```
int msb( int x )
```

che prende in ingresso un intero a 32 bit e restituisce la posizione dell'uno più significativo. Si assuma una codifica big-endian, in cui cioè la posizione dell'uno più significativo nella parola 0x80000000 è 31. Nel caso in cui il valore d'ingresso sia zero, la funzione deve restituire il valore -1.

Domanda B

Si consideri il seguente programma.

```
#include …

pthread_mutex_t  mutex = PTHREAD_MUTEX_INITIALIZER;
int              token = 2;
char             other;

void* threadA(void * arg)
{
    int varA = (int)arg;
    token = varA;
    pthread_mutex_lock(&mutex);
    varA--;
    token = varA;                                   /* breakpoint 1 */
    pthread_mutex_unlock(&mutex);
   return (void*)varA;
}

void* threadB(void * arg)
{
    int varB;
    varB = token;
    pthread_mutex_lock(&mutex);
    token  = varB;                                  /* breakpoint 2 */
    pthread_mutex_unlock (&mutex);
    return NULL;
}

int main()
{
    pthread_t ta, tb;
    int       counter = 7;
    pthread_create(&tb, NULL, threadB, NULL);
    pthread_create(&ta, NULL, threadA, (void *)counter );
    pthread_join (ta, NULL);
    pthread_join (tb, NULL);                       /*  breakpoint 3 */
}
```

Si completi la tabella seguente, riportando lo stato delle variabili secondo la classificazione:

- ESISTE — se la variabile certamente esiste
- NON ESISTE — se la variabile certamente non esiste
- PUO' ESISTERE — se la variabile potrebbe esistere oppure non esistere

Posizione	**varA**	**VarB**	**token**
Dopo breakpoint 1			
Dopo breakpoint 2			
Prima di breakpoint 3			
Dopo breakpoint 3			

Si completi la tabella seguente indicando tutti i possibili valori che le variabili riportate possono assumere nei punti specificati.

Posizione	**varA**	**VarB**	**token**
Dopo breakpoint 1			
Prima di breakpoint 2			
Dopo breakpoint 2			
Dopo breakpoint 3			

Domanda C

Si consideri un sistema con uno spazio di indirizzamento di 32 GByte ed una cache con le caratteristiche seguenti:

	D-Cache	I-Cache
Associatività	Set associativa a 8 vie	Diretta
Dimensione totale	512 KB	64 KB
Dimensione linea	256 B (per ogni set)	128 B
Tempo di accesso	2 ns	1 ns
Hit rate	99%	95 %

Si indichi la struttura dell'indirizzo visto dalle cache, descrivendo i vari campi e il loro significato.

Struttura dell'indirizzo D-Cache

Struttura dell'indirizzo I-Cache

Sapendo che:

- L’accesso alla memoria RAM avviene a parole di 64 bit
- Il tempo di accesso alla RAM in modalità normale è di 50 ns
- Il tempo di accesso alla RAM in modalità burst è
    - 80 ns per la prima parola
    - 20 ns per le parole successive

Si calcoli il tempo di accesso medio alla memoria attraverso le due cache

Tempo medio di accesso D-Cache

Tempo medio di accesso I-Cache

Domanda D

Dato il seguente programma:

```
1   int main()
2   {
3         FILE *f1, *f2,;
4         f1 = fopen(”/documents/mydoc/recent/newfile.del”,”w”);
5         f2 = fopen(”/home/reports/deliverables/del3.txt”,”w”);
6
7         fprintf(f2,”Deliverable n.”);
8
9         lseek(f1, -4, SEEK_CUR);
10        fprintf(f1,”place”);
11
12        fprintf(f2,”3”);
13        lseek(f2,-1,SEEK_CUR);
14
15        fclose(f2);
16        fclose(f1);
17  }
```

Ipotizzando che il file system sia organizzato come segue:

I-node list:	**Blocks:**
<0,dir,5> <3,dir,7> <9, dir,13> <10,dir,28> <13,dir,30> <14,dir,21> <33, norm, 70> <49, dir, 42> <56, norm, {84,85,86}>	005: …<3,documents>…<13,home> 007: …<9,mydoc > … 013: …<14, recent>… 021: …<33,newfile.del>… 028: …<49,deliverables>… 030: …<10,reports> … 042: …<56,del3.txt> 070: In a faraway land 084: Milan.

Sapendo che:

- L’i-node associato alla root directory “/” ha 0 come i-node number
- La i-list contiene terne del tipo: **<i-node-number, file-type, {block_list}>**
- Le directory contengono coppie del tipo: **< i-node-number, filename>**.
- La dimensione di un blocco è di 1024 byte
- Il sistema deve garantire che, per tutti i file aperti, il blocco contenente la posizione corrente sia in memoria principale
- Il buffer in memoria principale è preposto ad ospitare i blocchi ha dimensione 1024 B
- La scrittura dei file su disco avviene immediatamente (buffering a carattere)
- Per l’apertura del file è sempre necessario trasferire:
    - Un blocco per l’accesso all’i-node di ogni directory o file presente nel pathname, tranne che per la root directory
    - Un blocco per il contenuto di ogni directory presente nel pathname

Si completi la tabella seguente indicando con:

- **I<n>** l’accesso all’i-node n
- **B<k>** l’accesso al blocco k, se il blocco è su disco
- **M<k>** l’accesso al blocco k, se il blocco è in memoria

Chiamata di sistema	Posizione corrente	Sequenza accessi
`(4) f1 = open("...","w");`		
`(5) f2 = fopen("...","w");`		
`(7) fprintf(f2,"Deliverable n.");`		
`(9) lseek(f1, -4, SEEK_CUR);`		
`(10) fprintf(f1,"place")`		
`(12) fprintf(f2,"3");`		
`(13) lseek(f2,-1,SEEK_CUR)`		
`(15) fclose(f2);`		

Domanda E

Si sviluppi un programma multithreading in linguaggio C secondo le seguenti specifiche:

- Il programma dispone di un buffer condiviso della dimensione di un solo carattere
- Un thread “produttore” riceve una stringa come argomento e pone tale stringa un carattere alla volta nel buffer condiviso. La stringa è terminata dal carattere ‘\0’ che deve essere anch’esso copiato nel buffer. Il thread termina dopo aver copiato nel buffer il carattere terminatore '\0'.
- Un thread “consumatore” preleva un carattere alla volta dal buffer e lo stampa. Il thread termina quando preleva dal buffer il carattere terminatore ‘\0’. Il thread restituisce al programma principale la lunghezza della stringa stampata.
- Il programma attende la terminazione dei due thread e stampa la lunghezza della stringa.

Domanda F

Si disegni lo schema di un sistema a microprocessore dotato di tre periferiche connesse in daisy-chain. Si descriva quindi il funzionamento del daisy-chain ed il meccanismo di interrupt vettorizzato.