Politecnico di Milano – Sede di Cremona
Anno Accademico 2022/2023

Architettura dei Calcolatori e Sistemi Operativi

Esame – 08.06.2023

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

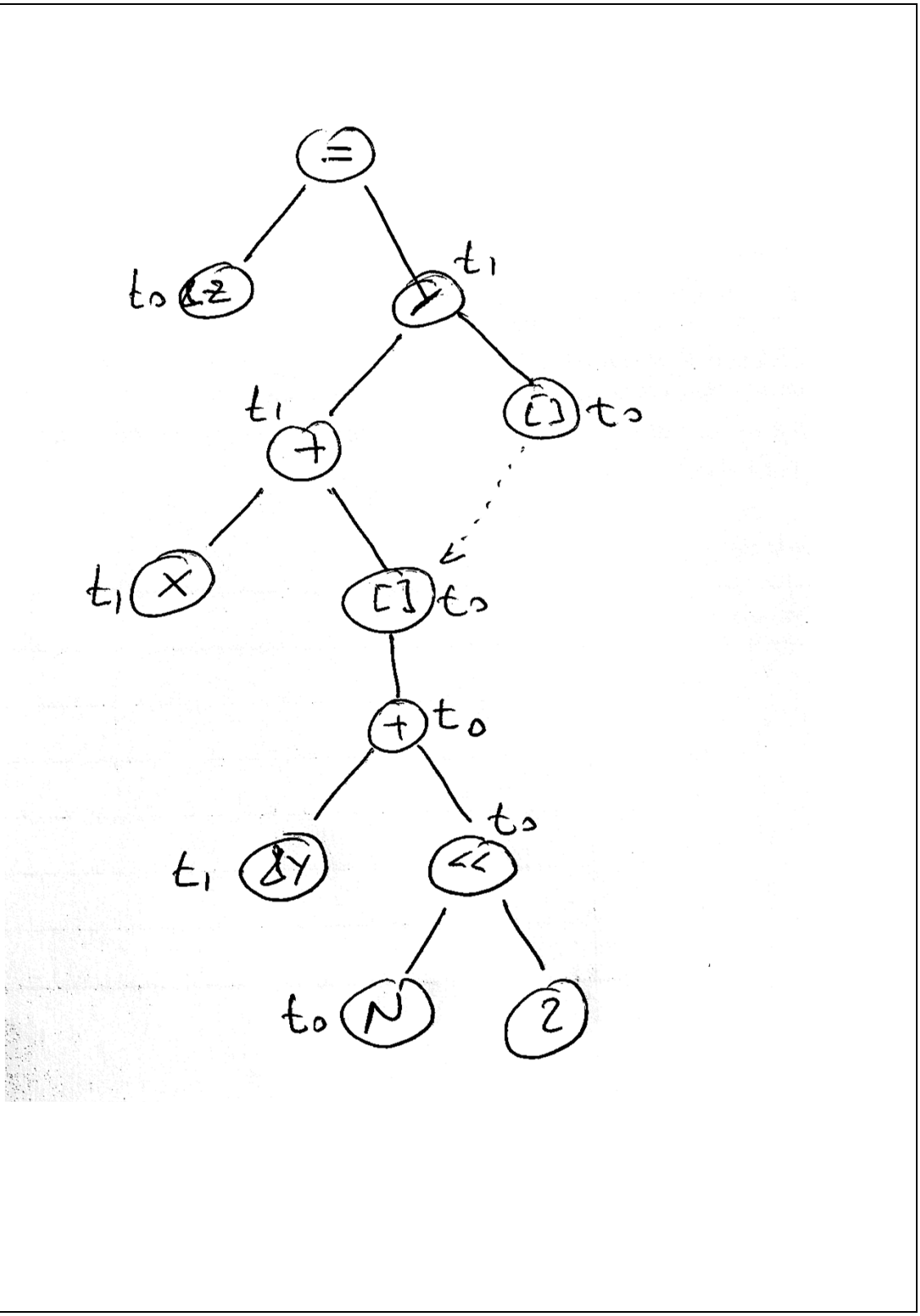
Domanda	Voto	Note
A		
B		
C		
D		
E		
F		

Domanda A

Si consideri la seguente espressione in C, in cui i tipi di tutte le variabili e gli array è intero a 32 bit con segno:

```
Z = (X + Y[N]) / Y[N];
```

Si disegni l'albero sintattico completo relativo a tale espressione e si proceda alla allocazione delle variabili e dei risultati intermedia ai registri temporanei in modo da minimizzare il numero di registri utilizzati. Si annoti l'albero con l'indicazione dei registri utilizzati nei diversi punti.



Sulla base dell'albero sintattico ottenuto, si traduca l'espressione in linguaggio assembly.

```
la      $t0, N              # t0 = &N
lw      $t0, ($t0)          # t0 = N
sll     $t0, $t0, 2         # t0 = 4N
la      $t1, Y              # t1 = &Y
add     $t0, $t0, $t1       # t0 = &Y + 4N = &(Y[N])
lw      $t0, ($t0)          # t0 = Y[N]
la      $t1, X              # t1 = &X
lw      $t1, ($t1)          # t1 = X
add     $t1, $t1, $t0       # t1 = X + Y[N]
div     $t1, $t0            # hi, lo = (X + Y[N]) / Y[N]
mflo    $t1                 # t1 = (X + Y[N]) / Y[N]
la      $t0, Z              # t0 = &Z
sw      $t1, ($t0)
```

Domanda B

Si realizzi un programma in C secondo la seguente specifica e considerando lo stralcio di codice riportato nel seguito.

1. Il programma utilizza un vettore globale **`values`** di 5000 interi
2. Il programma deve elaborare l'intero vettore massimizzando il parallelismo
3. Il programma genera 5 processi figli
4. Ogni processo figlio elabora 1000 elementi del vettore ed in particolare
    a. Stampa l'indice del primo e dell'ultimo elemento che elabora
    b. Calcola il totale degli elementi
    c. Stampa il totale degli elementi
5. Quando tutti i figli sono terminati il programma stampa un messaggio di fine calcolo.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

int values[5000] = { ... };

int main()
{
   pid_t pid;

   for( int i = 0; i < 5; i++ )
   {
      pid = fork();
      if( pid == 0 )
      {
         // CHILD
         int tot = 0;
         for( int j = 1000*i; j < 1000*(i+1); j++ )
         {
            tot += values[j];
         }
         printf( "Start: %d\n", 1000*i )
         printf( "End:   %d\n", 1000*i+999 )
         printf( "Total: %d\n", tot )
         exit( 0 );
      }
    }

    // PARENT
    for( int i = 0; i < 5; i++ )
    {
       wait(NULL);
    }
    printf( "Done\n");
    exit( 0 );
}
```

Domanda C

Si consideri un sistema con uno spazio di indirizzamento di 1 GByte ed una memoria cache con le caratteristiche seguenti:

Associatività	Set associativa a 4 vie
Dimensione totale	128 KB
Dimensione linea	256 B (per ogni via)
Tempo di accesso	1 ns
Hit rate	99.5 %

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

Struttura dell'indirizzo

OFFSET = log2(256) = 8 bit

INDEX = log2(128K / (256 * 4)) = 7 bit

TAG = log2(1G) – 8 – 7 = 15 bit

Sapendo che:

- L'accesso alla memoria RAM avviene a parole di 128 bit
- Il tempo di accesso alla RAM in modalità normale è di 90 ns
- Il tempo di accesso alla RAM in modalità burst è
    - 180 ns per la prima parola
    - 40 ns per le parole successive

Si calcoli il tempo di accesso medio alla memoria.

Tempo medio di accesso

Tcache = 0.995 * 1ns + 0.005 * (1 + 180 + ((256 / 16) – 1) * 40) = 4.9 ns

Supponendo che il 25% delle istruzioni eseguite da un certo programma siano di load/store, si calcoli il tempo di accesso medio per istruzione con e senza cache ed il miglioramento delle prestazioni.

Tempo di accesso medio con cache:

Tave = (100* Tcache + 25 * Tcache) / 100 = 6.125 ns

Tempo di accesso medio senza cache:

Tave = (100* Tram + 25 * Tram) / 100 = 112.5 ns

Miglioramento delle prestazioni:

R = (112.5 – 6.125) / 112.5 = 94.5%

Domanda D

Si consideri un calcolatore dotato di sistema operativo Linux e in cui valgono le seguenti specifiche:

- La dimensione dei blocchi è di 512 byte
- Per l'apertura dei file è sempre necessario accedere a:
  - i-node di ogni cartella o file presente nel percorso
  - Blocco per il contenuto di ogni cartella presente nel percorso
  - Primo blocco dati del file

Dato il contenuto del seguente volume:

i-node list: <0, dir, 10><1, dir, 11><2, dir, 12><3, norm, {100, 101}>
<4, dir, 13><5, dir, 14><6, norm, {200, 201, 202, 203, 204, 205}>
Blocco 10: ...<1, home><2, bin><4, tmp>
Blocco 11: ...<7, documents><8, images>
Blocco 12: ...<9, bash><10, ls>
Blocco 13: ...<3, foo.txt><5, acso>
Blocco 14: ...<6, esame.txt>
...
Blocco 100: dati
Blocco 101: dati
...
Blocco 200: dati
...
Blocco 205: dati

Si consideri inoltre il programma seguente:

```
int main()
{
   int fd;
   char buf[1024];

   fd = open( "/tmp/acso/esame.txt", O_RDONLY );
   if( fork() == 0 ) {
       lseek( fd, 1024, SEEK_SET );
       read( fd, buf, 700 );
       close( fd );
       exit( 0 );
   }

   wait( NULL );
   write( fd, buf, 900 );
   close( fd );

   exit( 0 );
}
```

Per ciascuna delle seguenti chiamate di sistema, si completi la tabella riportando la sequenza di accesso agli i-node ed ai blocchi in memoria principale (M) o su disco (D). **Si tenga presente che ogni operazione di scrittura richieda sempre l'accesso all'intero blocco su disco.**

Chiamata di sistema	**Sequenza di accessi**
fd = open(...)	I-node 0 (D) -> Blocco 10 (D) I-node 4 (D) -> Blocco 13 (D) I-node 5 (D) -> Blocco 14 (D) I-node 6 (D) -> Blocco 200 (D)
lseek(...)	Nessun accesso / oppure I-node 6 (M)
read(...)	I-node 6 (M) -> Blocco 202 (D), Blocco 203 (D)
close(...) – processo figlio	Nessun accesso
write(...)	I-node 6 (M) -> Blocco 200 (D), Blocco 201 (D)
close(...) – processo padre	Nessun accesso

Domanda E

Si consideri architettura a 32 bit dotata di un sistema operativo con memoria paginata e segmentata, sul quale venga compilato il seguente programma:

```
#define MSG "Hello World!"

typedef struct {
  int  A;
  char B;
  int  C;
} record_t;

int               X;
static const int  D = 3;
char              W[64] = {};
record_t          S1;
record_t          S2 = { 10, 20, 30 };

int main()
{
  char* SPTR = MSG;
  ...
}
```

Si compili la seguente tabella per ognuno dei simboli presenti nel programma secondo le seguenti indicazioni:

- Tipo C. Indica l'espressione C che esprime il tipo del simbolo
- Segmento. Indica il segmento in cui il simbolo sarà allocato
- Dimensione ELF. Indica la dimensione occupata dal simbolo nel file ELF
- Dimensione RAM. Indica la dimensione occupata dal simbolo una volta caricato in RAM

Simbolo	Tipo C	Segmento	Dimensione ELF	Dimensione RAM
MSG	macro	//	//	//
"Hello world!"	char*	rodata	13	13
X	int	bss	0	4
D	int	rodata	4	4
		text	4	4
		//	//	0
W	char	data	64	64
		bss	0	64
S1	struct	bss	0	12
S2	struct	data	12	12
main	int (*f)()	text	0	0
SPTR	char*	stack	0	4

Per i simboli D e W tutte le soluzioni proposte sono corrette.

Domanda F

Si disegni il diagramma semplificato (5 stati) dello stato dei processi e si descriva in modo sintetico e preciso:

- Il significato di ogni stato
- Il significato di ogni transizione