



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

Architettura dei Calcolatori e Sistemi Operativi

Esame – 18.01.2022

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 range( int* array, int size )
```

che prende in ingresso un vettore array di dimensione size e restituisce come valore di ritorno il range degli elementi ovvero la differenza tra il massimo ed il minimo del vettore. Il codice che segue è un esempio di come un programma dovrebbe utilizzare la funzione.

```
        .data
array:  .word 10, 13, ...
size:   .word 110

        .text
main:
        la     $a0, array
        lw     $a1, size
        jal   range
        move   $a0, $v0
        li     $v0, 1
        syscall
```

Domanda B

Si consideri il seguente codice assembly MIPS:

```
LW    $s0, 4($t0)
ADDI  $s0, $s0, 1
ADDI  $t1, $zero, 3
ADDI  $t2, $zero, 4
XOR   $s1, $s1, $s1
XOR   $s1, $t1, $t2
```

Si supponga di eseguire il codice su un processore MIPS **privo di percorsi di forwarding e incapace di introdurre stalli** hardware in modo autonomo. Si modifichi il codice dato inserendo le istruzioni NOP necessarie a garantire una corretta esecuzione.

Si ottimizzi quindi il codice ottenuto riordinando le istruzioni rimuovendo le eventuali istruzioni NOP che non sono più necessarie.

Si supponga ora di eseguire il nuovamente il codice su un processore MIPS **dotato di percorsi di forwarding** e ma **incapace di introdurre stalli** hardware in modo autonomo. Si modifichi il codice inserendo le istruzioni NOP necessarie a garantire la corretta esecuzione.

Infine, si ottimizzi il codice appena ottenuto riordinando le istruzioni rimuovendo le eventuali istruzioni NOP che non sono più necessarie.

Domanda C

Si consideri un sistema con uno spazio di indirizzamento di 256MByte ed una cache con le caratteristiche seguenti:

Associatività	Set associativa a 4 vie
Dimensione totale	128 KB
Dimensione linea	128 B (per ogni set)
Tempo di accesso	1 ns
Hit rate	99%

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

Struttura dell'indirizzo

Sapendo che:

- L'accesso alla memoria RAM avviene a parole di 64 bit
- Il tempo di accesso alla RAM in modalità normale è di 40 ns
- Il tempo di accesso alla RAM in modalità burst è
 - 50 ns per la prima parola
 - 20 ns per le parole successive

Si calcoli il tempo di accesso medio alla memoria.

Tempo medio di accesso

Domanda D

Si consideri il seguente insieme di processi:

Process	Arrival Time (T_A)	Execution Time (T_E)
P1	0	5
P2	1	5
P3	2	2
P4	10	8
P5	12	4

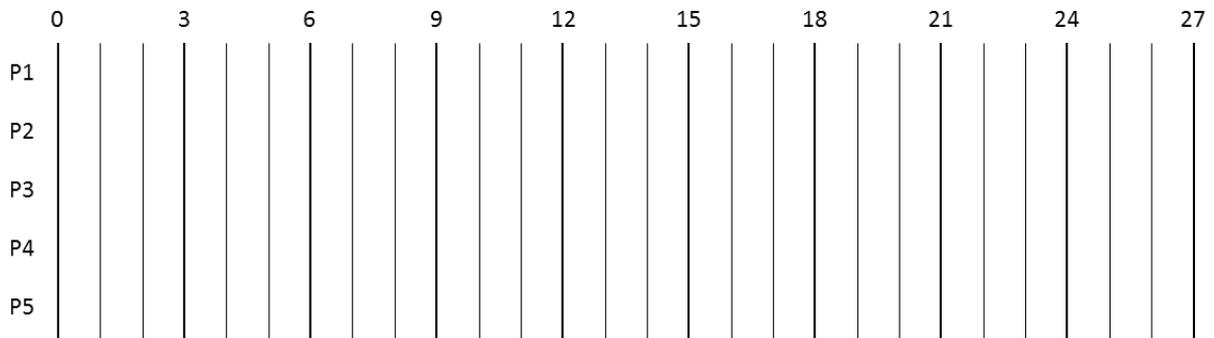
Si esegua lo scheduling di tali processi secondo i due seguenti algoritmi:

- Round Robin, non-preemptive, con un quanto di tempo pari a 3 unità
- Shortest Remaining Time

Per ognuno dei due casi, quindi, si svolgano i seguenti punti:

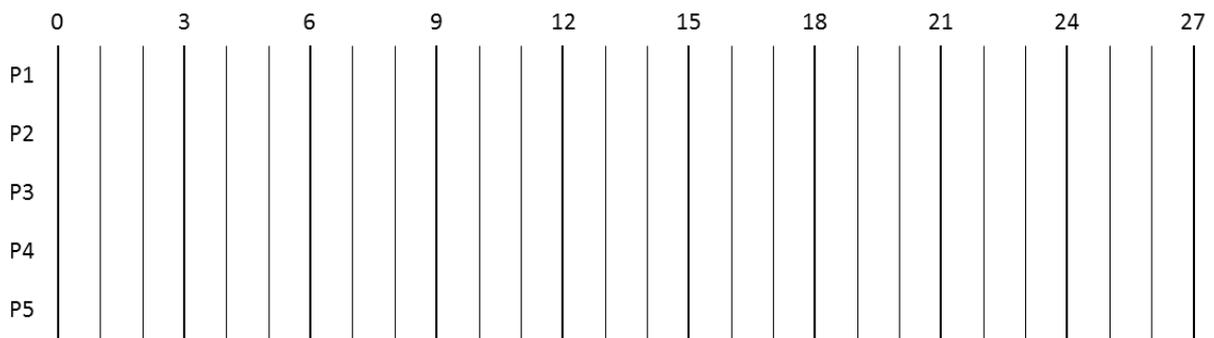
- Si indichi il tempo reale di esecuzione di ogni processo
- Si calcoli il tempo di attesa medio T_w dei processi

Round Robin



$T_w =$

Shortest Remaining Time



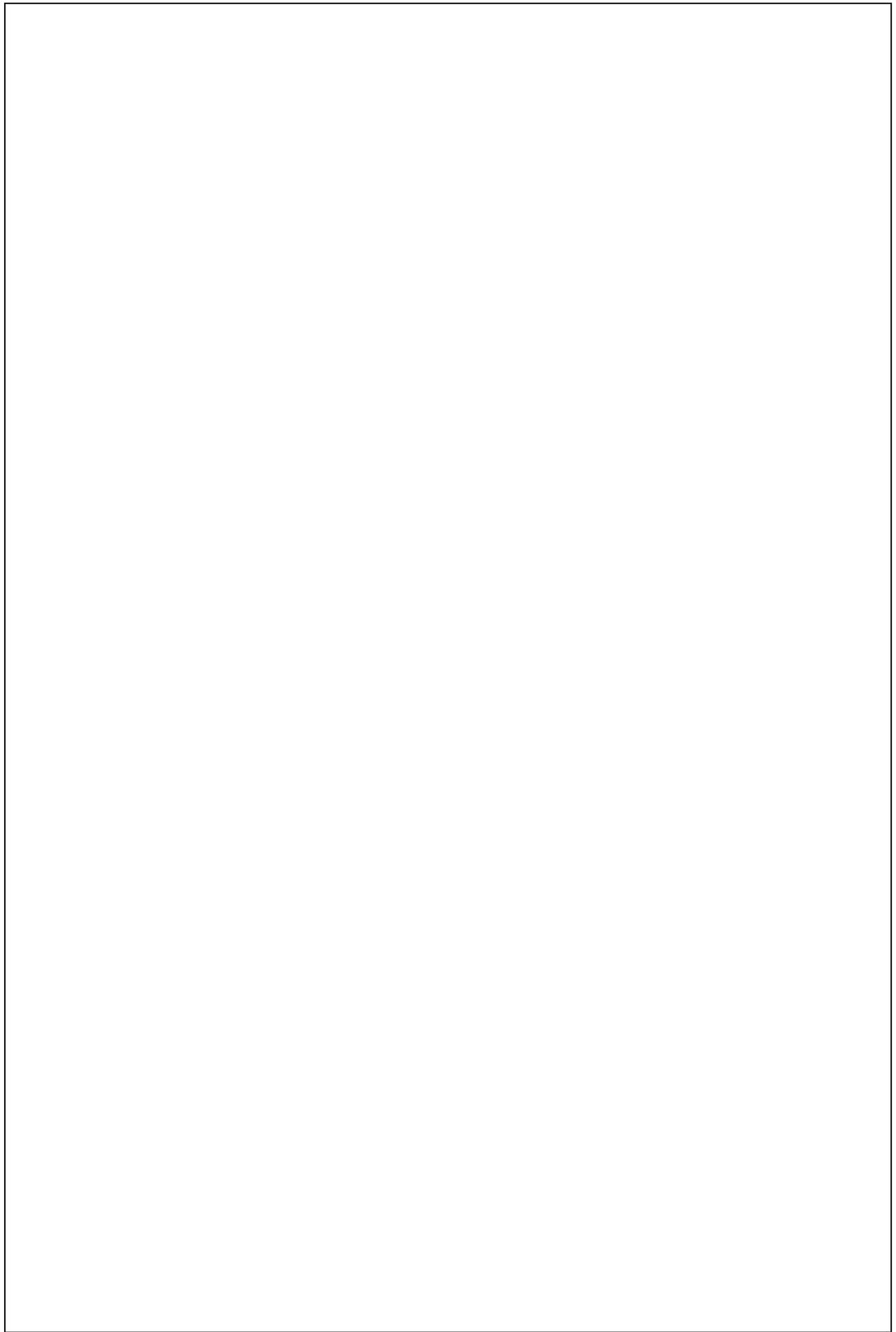
$T_w =$

Domanda E

Si consideri il seguente gioco a due giocatori. Ogni giocatore, a turno, lancia due dadi. Il gioco continua fino a che un giocatore ottiene un punteggio maggiore o uguale a quello che l'altro giocatore ha ottenuto al turno precedente. Quando il valore ottenuto da un giocatore è inferiore a quello del precedente turno, il giocatore perde e l'avversario vince.

Si scriva un programma che simuli il gioco, secondo le seguenti specifiche:

- Un giocatore è realizzato mediante un thread
- Il thread viene creato passando il nome del giocatore come argomento
- Ogni thread deve stampare il nome del giocatore corrente ed il punteggio ottenuto
- Infine il programma deve stampare il vincitore.



Domanda F

Si descriva il tipico flusso di compilazione, indicando chiaramente, per ogni fase, la natura dei dati (file) in ingresso ed in uscita e le operazioni svolte.