



Politecnico di Milano – Sede di Cremona
Anno Accademico 2017/2018

Architettura dei Calcolatori e Sistemi Operativi

Esame – 02.07.2018

Prof. Carlo Brandoles

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 funzione delle librerie standard del C:

```
void* memcpy( void* dst, const void* src, size_t n )
```

che copia n byte dalla zona di memoria puntata da src a quella puntata da dst e restituisce un puntatore all'area di destinazione. La specifica della funzione indica inoltre che se le due zone di memoria si sovrappongono, anche solo parzialmente, la copia non viene effettuata e la funzione restituisce un puntatore nullo. Come ulteriore indicazione si consideri il codice seguente che mostra un utilizzo corretto della funzione in esame.

```
.data
dfrom: .space 128
str:   .ascii "some stuff\n"
dto:   .space 32
.text
main:
    la    $a0,dto
    la    $a1,dfrom
    li    $a2,16
    jal   memcpy
    beq  $v0, $zero, error
    ...
```

Si traduca la funzione `memcpy()` in assembly.

Domanda B

Si supponga che il codice riportato di seguito sia prodotto da un compilatore la cui architettura target è un processore MIPS senza ottimizzazioni. Per ogni istruzione si indentifichi:

- La tipologia di conflitto che viene generato (DATI, CONTROLLO, N/A)
 - Il numero di cicli di stallo che è necessario inserire dopo l'istruzione per garantire il corretto funzionamento del programma

ISTRUZIONE	TIPO DI CONFLITTO	NUMERO DI STALLI
ADD R0, R1, R2		
LOOP: SUB R1, R0, R3		
MUL R2, R0, R3		
LW R4, 0(R5)		
ADDI R4, R4, 4		
SUB R5, R6, R7		
ADDI R5, R5, 1		
BEQ R0, R6, LOOP		
MUL R8, R9, R10		
ADDI R9, R9, 5		

Si supponga che l'architettura target non disponga dell'unità di rilevamento dei conflitti. Nella tabella BASE, si riporti quindi il codice modificato inserendo le istruzioni NOP necessarie per far sì che l'architettura possa eseguirlo correttamente. Si supponga infine che il compilatore possa ottimizzare il codice riordinando le istruzioni per migliorare i tempi di esecuzione. Si riscriva il codice opportunamente modificato nella tabella OTTIMIZZATO.

Domanda C

Si consideri un sistema con uno spazio di indirizzamento di 1 GByte, una cache dati ed una cache istruzioni. Le caratteristiche delle due cache sono le seguenti:

	Cache Istruzioni	Cache Dati
Associatività	Mapping diretto	Set associativa a 8 vie
Dimensione totale	64 KB	256 KB
Dimensione linea	128 B	256 B (per ogni set)
Tempo di accesso	1 ns	2 ns
Hit rate	95%	98%

Sulla base di queste informazioni si indichi la struttura dell'indirizzo visto dalle cache, descrivendo i vari campi e il loro significato.

Cache Istruzioni

Cache Dati

Sapendo che:

- L'accesso alla memoria RAM avviene a parole di 32 bit
- Il tempo di accesso alla RAM in modalità normale è di 50 ns
- Il tempo di accesso alla RAM in modalità burst è
 - 60 ns per la prima parola
 - 15 ns per le parole successive

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

Cache Istruzioni

Cache Dati

Considerando un programma in cui:

- il 10% delle istruzioni sono mispredicted branch e richiedono uno stallo
- il 30% delle istruzioni sono load/store
- le restanti istruzioni sono aritmetiche

Si calcoli il tempo medio di esecuzione di una istruzione con e senza cache.

Tempo medio per istruzione senza cache

Tempo medio per istruzione con cache

Domanda D

Si scriva un programma `pprimes` che accetta due parametri da linea di comando:

```
$> pprimes N P
```

Il programma:

- Calcola in parallelo su N thread quali numeri da 1 a P sono primi
- Stampa a video in ordine crescente i numeri primi identificati

Alcuni controlli opportuni sugli ingressi sono:

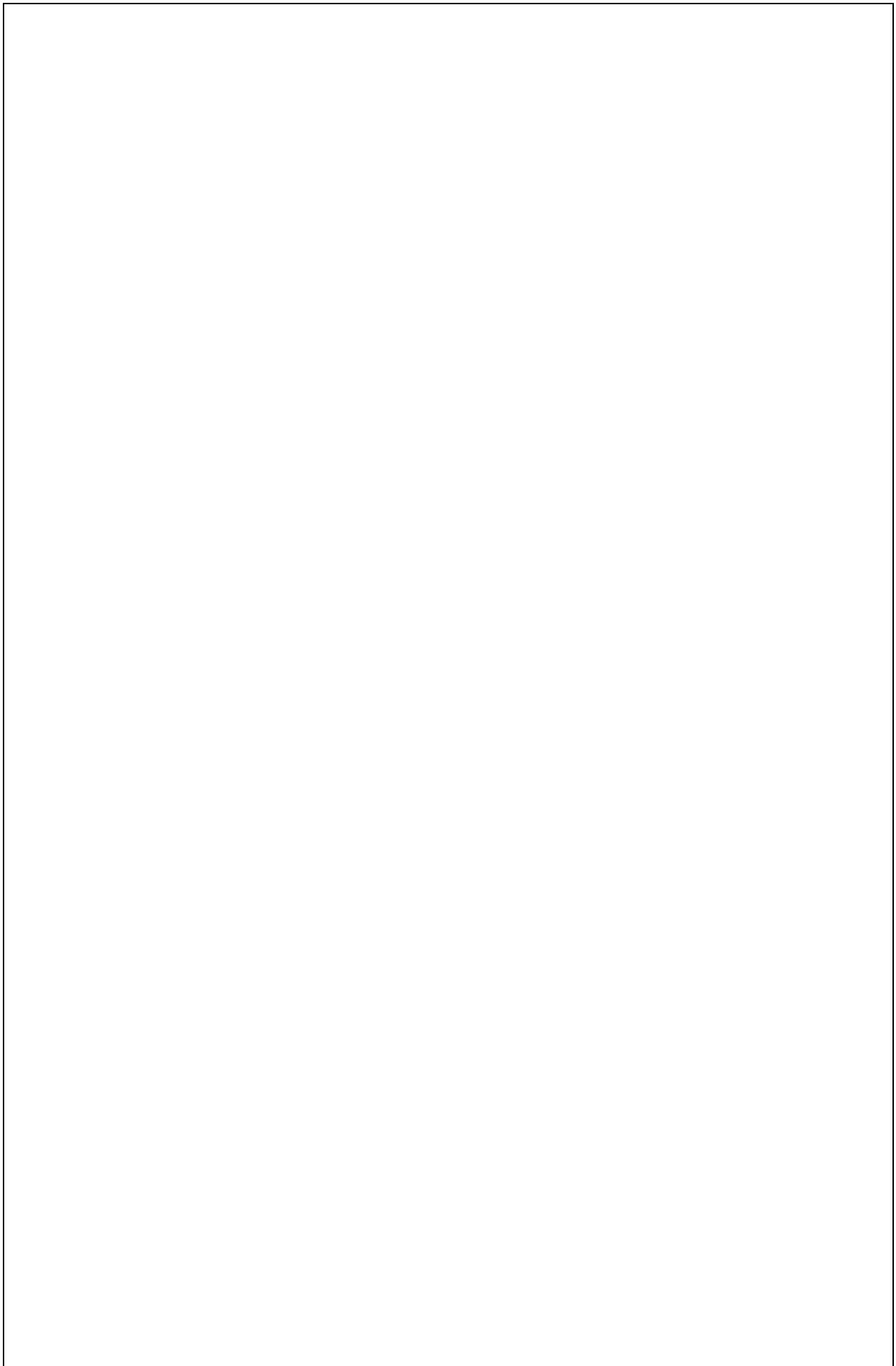
- $N > 1$
- $1 < P < 10000$

Secondo le specifiche sopra riportate la seguente esecuzione

```
pprimes 3 10
```

istanzia 3 thread per calcolare i numeri primi da 2 a 10 in parallelo e darà come output a video:

```
2 3 5 7
```



Domanda E

Si consideri un sistema dotato di memoria virtuale con paginazione e segmentazione di tipo UNIX caratterizzato dai parametri seguenti:

Memoria logica: 32 Kbyte
Memoria fisica: 16 Kbyte
Dimensione pagina: 4 Kbyte

Si definisca la struttura degli indirizzi fisico e logico indicando la lunghezza dei campi NPF, Spiazzamento fisico, NPL, Spiazzamento logico.

Indirizzo fisico:	
Indirizzo logico:	

Si considerino due programmi X e Y caratterizzati dalla seguente dimensione iniziale dei segmenti (C: Code, D: Data, P: Pila)

CX: 4K DX: 12K PX: 4K
CY: 8K DY: 4K PY: 4K

Completere la seguente tabella, riportando la struttura in pagine della memoria virtuale dei due programmi X e Y sapendo che le pagine vengono allocate sequenzialmente.

Inizia quindi l'esecuzione e ad un certo istante T le seguenti operazioni sono state completate:

- Il processo P viene creato ed esegue il programma Y
 - Lo stack del processo P cresce ed alloca una nuova pagina
 - Il processo Q viene creato ed esegue il programma X
 - Il processo P salta all'indirizzo 0x05BC
 - Il processo Q accede ad un dato all'indirizzo 0x2EA0
 - Il processo P fa una fork e crea un nuovo processo R
 - Il processo R accede ad un dato all'indirizzo 0x2484
 - Lo stack del processo Q cresce di una pagina
 - Il processo R termina

Sapendo che:

1. L'esecuzione di un programma avviene caricando inizialmente e in quest'ordine:
 - a. La pagina di codice con l'istruzione di partenza
 - b. Una pagina di pila
 2. Il caricamento di ulteriori pagine in memoria avviene su richiesta (on demand)
 3. Il numero di pagine residenti di R è pari a 3
 4. L'indirizzo esadecimale di partenza di X è 0x0124
 5. L'indirizzo esadecimale di partenza di Y è 0x1A64
 6. Per la sostituzione delle pagine di memoria si utilizza una politica LRU. Almeno una pagina di pila deve sempre rimanere in memoria.
 7. L'allocazione delle pagine virtuali nelle pagine fisiche avviene sempre in sequenza
 8. All'inizio della sequenza di eventi la MMU è vuota

Si riporti nelle tabelle seguenti una descrizione dello stato della memoria fisica e della MMU al termine della sequenza di operazioni di cui sopra.

Domanda F

Si descriva il concetto di “stato” di un processo, riportando e commentando il diagramma di transizione. Si indichino in particolare le condizioni sotto le quali avvengono le transizioni e quali sono gli scheduler che ne hanno in carico la gestione.