

## Programmazione di sistema

<b>Iniziato</b>	martedì, 18 ottobre 2022, 15:04
<b>Stato</b>	Completato
<b>Terminato</b>	martedì, 18 ottobre 2022, 15:04
<b>Tempo impiegato</b>	8 secondi
<b>Valutazione</b>	<b>0,00</b> su un massimo di 15,00 ( <b>0%</b> )

**Domanda 1**

Risposta non data

Punteggio max.:

3,00

**ATTENZIONE: E' OBBLIGATORIO CONNETTERSI ALLA VIRTUAL CLASSROOM DI SUPPORTO CON LA WEBCAM ACCESA**

**SE I RISULTATI SONO NUMERI, RIPORTARE PASSAGGI INTERMEDI RILEVANTI E/O FORMULE USATE**

**LE RISPOSTE SI/NO VANNO MOTIVATE**

Sia dato un sistema di memoria virtuale con paginazione, nel quale vengono indirizzati i Byte. Il sistema dispone di TLB (Translation Look-aside Buffer), su cui si misura sperimentalmente un "hit ratio"  $H$ . La tabella delle pagine ("page-table") viene realizzata con uno schema a due livelli, nel quale un indirizzo logico di 64 bit viene suddiviso (da MSB a LSB) in 3 parti:  $p1$ ,  $p2$  e  $d$ , rispettivamente di 40 bit, 12 bit e 12 bit. Non si utilizzano ulteriori strutture dati (quali tabelle di hash o inverted page table) per velocizzare gli accessi. La memoria virtuale viene gestita con paginazione a richiesta.

Si risponda alle seguenti domande:

- A) Supponendo che la memoria RAM abbia tempo di accesso di  $T_{RAM} = 200$  ns, si calcoli  $H$  tale da garantire tempo effettivo di accesso (EAT) minore di 250 ns, assumendo che il tempo di accesso alla TLB sia trascurabile.
- B) Si consideri ora la frequenza di page fault  $p=10^{-5}$ . in quanto tempo  $T$  viene servito un un page fault se si osserva che il degrado nel tempo di accesso dovuto ai soli PF è superiore al 10%? Si tratta di un valore massimo o minimo per  $T$ ? (Siccome  $H$  non è noto, ma è dato da un intervallo di valori, si calcoli  $T$  per  $H$  minimo e massimo)
- C) Data una stringa di riferimenti a indirizzi in memoria (letture e scritture), il numero di page fault può variare al variare della dimensione di una pagina (motivare)? A parità di tutto il resto, può servire modificare la dimensione (numero di bit) di  $p1$  e  $p2$  (senza cambiare  $d$ ) per ridurre il numero di PF (motivare)?

A) Calcolo  $H$  :

$T_{RAM} = 200$  ns (RAM access time)

PT a 2 livelli (gerarchica) => 2 letture per ogni PT lookup

$$EAT_{PT} = H \cdot T_{RAM} + (1-H) \cdot 3T_{RAM} = (3-2H)T_{RAM}$$

$$EAT_{PT} < 250\text{ns}$$

$$(3-2H)200\text{ns} < 250\text{ns}$$

$$3-2H < 1.25$$

$$H > 1.75/2 = 0.875 = 87.5\%$$

B) Calcolo  $T$  che causa un degrado  $> 20\%$  per EAT.

Si tratta di un valore massimo o minimo per  $T$ ?

T: PF service time

$EAT_{PT}$ : Effective Access Time dovuto a Page Table e TLB

$EAT_{PT+PF}$ : Effective Access Time dovuto a Page Table (con TLB) e Page Fault

$EAT_{PT+PF} = (1-p) \cdot EAT_{PT} + p \cdot T = EAT_{PT} + p (T - EAT_{PT}) = EAT_{PT} + p \cdot T$  //  $EAT_{PT}$  è trascurabile rispetto a T

$EAT_{PT+PF} > 1.1 EAT_{PT}$

$EAT_{PT} + p \cdot T > 1.3 EAT_{PT}$

$T > (0.1 EAT_{PT}) / p = 10^4 EAT_{PT}$

$H_{min} = 0,875) EAT_{PT} = 250 \text{ ns} \Rightarrow T > 2.5 \text{ ms}$

$H_{MAX} = 1) EAT_{PT} = 250 \text{ ns} \Rightarrow T > 2.0 \text{ ms}$

Valor minimo o mimino (dire perché) ? Valore minimo, vista la disequazione, oppure osservando che al diminuire di T diminuirebbe il tempo di accesso (per il quale si da invece un valore minimo oltre cui non si può scendere)

C) Data una stringa di riferimenti a indirizzi in memoria (letture e scritture), il numero di page fault può variare al variare della dimensione di una pagina (motivare)? A parità di tutto il resto, può servire modificare la dimensione (numero di bit) di p1 e p2 (senza cambiare d) per ridurre il numero di PF (motivare)?

1) La dimensione della pagina influisce sui PF?

SI. Influisce in quanto diminuisce il numero di pagine e, a parità di riferimenti, può favorire la località: il caso estremo sarebbe quello di una sola pagina molto grande.

2) il numero di bit di p1 e p2 influisce sui PF?

NO. Può solo influire sull'organizzazione della PT gerarchica, ma non sul numero di PF, che resta invariato in quanto, con d costante, la dimensione della pagina non cambia.

**Domanda 2**

Risposta non data

Punteggio max.:

3,00

**TUTTE LE RISPOSTE SÌ / NO DEVONO ESSERE MOTIVATE. QUANDO I RISULTATI SONO NUMERI, SONO NECESSARI IL RISULTATO FINALE E I RELATIVI PASSI INTERMEDI (O FORMULE)**

Considerare un file system Unix-like, basato su inode, con 13 puntatori / indici (10 diretti, 1 singolo indiretto, 1 doppio indiretto e 1 triplo indiretto). I puntatori / indici hanno una dimensione di 32 bit e i blocchi del disco hanno una dimensione di X Byte. Il file system risiede su una partizione del disco di 800 GB, che **include sia blocchi di dato che blocchi di indice**.

A) Supponendo che tutti i metadati (eccetto i blocchi indice) abbiano dimensioni trascurabili, calcolare i possibili valori di X, tenendo conto che si tratta di una potenza di 2 e che il file system deve garantire di poter contenere fino a 100 file con indicizzazione indiretta tripla. Si calcolino i valori minimo e massimo per X

B) Dato un file binario di dimensione 10381.25 KB, calcolare esattamente quanti blocchi indice e blocchi di dato occupa il file, corrispondenti ai valori minimo e massimo precedentemente calcolati per X

---

A) Supponendo che tutti i metadati (eccetto i blocchi indice) abbiano dimensioni trascurabili, calcolare il numero massimo di file che il file system può ospitare, utilizzando l'indicizzazione indiretta singola (N1) e l'indicizzazione indiretta tripla (N3).

### **Osservazioni generali**

*Un blocco indice contiene  $x / 4$  puntatori / indici.*

*La partizione contiene  $800G / X = 800/X$  M blocchi*

**Per garantire che 100 file con indice indiretto triplo occorre considerare la dimensione massima MAX3**

$$MAX3 = 10 + X/4 + (X/4)^2 + (X/4)^3 \text{ data blocks} + \\ 1 \text{ (single)} + 1 + X/4 \text{ (double)} + 1 + X/4 + (X/4)^2 \text{ (triple) index blocks}$$

**Benché si potrebbe risolvere l'esercizio con disequazioni e formula complete., una soluzione molto più rapida si ottiene per semplici tentativi con dati approssimati**

$$MAX3 = (X/4)^3 \text{ data blocks}$$

$$A) X = 1K, X/4 = 256 = 2^8$$

$$MAX3 = 2^{24} \text{ blocks} = 2^{34} \text{ Bytes} = 16GB$$

$$100 * MAX3 = 1600GB > 800GB - \text{quindi TROPPO}$$

$$B) X = 512, X/4 = 128 = 2^7$$

$$MAX3 = 2^{21} \text{ blocks} = 2^{30} \text{ Bytes} = 1GB.$$

$$100 * MAX3 = 100GB < 800GB - \text{Va bene e l'approssimazione è abbondantemente corretta}$$

$$\text{Risposta: } X = 512 - \text{block size} = 512 \text{ Bytes}$$

### **Valor Minimo**

Qualunque dimensione di blocco inferiore a 512 B è compatibile con il vincolo. Si potrebbe considerare come limite minimo (escluso) 0, oppure una dimensione tale da contenere almeno 2 o 4 indici, ma sarebbero in ogni caso dimensioni poco sensate (ci sono altre considerazioni che sconsigliano blocchi troppo piccoli). In sostanza la risposta importante in questo problema è il limite massimo.

- B) Dato un file binario di dimensione 10381.25 KB, calcolare esattamente quanti blocchi indice e blocchi di dato occupa il file, corrispondenti ai valori minimo e massimo precedentemente calcolati per X

$$\text{Data blocks: } \text{ceil}(10381.25KB/0.5KB) = 20763$$

Index blocks

Single index: 1

$$\text{Inner index blocks (double): } 128 \text{ (il double indirect copre fino a } 10+128+128^2 = 16522 \text{ data blocks)}$$

Outer index block (double): 1

$$\text{Inner index blocks (triple): } \text{ceil}((20763-16522)/128) = 34$$

Outer index block (triple): 1 + 1

$$\text{Total index blocks: } 1+128+1+34+1+1 = 166$$



**Domanda 3**

Risposta non data

Punteggio max.:

3,00

**TUTTE LE RISPOSTE SÌ / NO VANNO MOTIVATE. PER LE RISPOSTE NUMERICHE, SONO RICHIESTI SIA I RISULTATI CHE I PASSAGGI INTERMEDI (O FORMULE) RILEVANTI**

Si risponda alle seguenti domande sulla gestione della memoria:

- A) Si consideri il caricamento dinamico (dynamic loading) e il link dinamico (dynamic linking). È possibile caricare dinamicamente un programma senza che sia necessario il dynamic linking? Il dynamic linking richiede che un programma sia anche caricabile dinamicamente (dynamic loading)?
- B) Si spieghi brevemente perché un'Inverted Page Table necessita di una tabella di HASH. Perché la soluzione di IPT + tabella di HASH è diversa da una soluzione con PT basata solamente su tabella di Hash?
- C) Si consideri una CPU dotata di TLB: la TLB può contenere entry di più processi simultaneamente o è vincolata a contenere entry per un solo processo?
- 

- A) È possibile caricare dinamicamente (dynamic loading) un programma senza che sia necessario il dynamic linking?

Sì. Sebbene il link dinamico possa essere combinato al caricamento (load) dinamico, non è obbligatorio: un programma può essere linkato staticamente e caricato in modo dinamico/incrementale, ad es. caricamento dinamico controllato dal programma (program-based dynamic loading).

*Nota: caricamento (load) dinamico significa che i pezzi di un programma vengono caricati in memoria solo se/quando necessari, il link dinamico significa che i riferimenti tra moduli vengono risolti in fase di esecuzione (è particolarmente utile per le librerie condivise/shared).*

Il dynamic linking richiede che un programma sia anche caricabile dinamicamente (dynamic loading)?

No. Ancora una volta, sebbene il caricamento dinamico possa sfruttare il link dinamico, un programma può essere linkato dinamicamente (ad esempio a una libreria condivisa già caricata in memoria) senza caricamento (load) dinamico

- B) Si spieghi brevemente perché un'Inverted Page Table necessita di un campo pid (ID del processo) in ciascuna delle sue entry, mentre ciò non è vero per una tabella di pagine standard.

Perché se non si usasse una tabella di HASH occorrerebbe una ricerca lineare del numero di pagina logica (p) nella IPT.

Le due soluzioni differiscono perché, pur essendo molto simili in termini di prestazioni della tabella di HASH, in un caso (con la IPT, vengono inseriti direttamente nelle liste di chaining della tabella di HASH le entry della IPT, mentre nel caso si HASH semplice occorrono le classiche allocazioni e deallocazioni di elementi ad ogni inserimento/cancellazione dalla HASH. Quindi soluzione IPT+HASH è più efficiente nell'uso della memoria.

C) Si consideri una CPU dotata di TLB: la TLB può contenere entry di più processi o è vincolata a contenere entry per un solo processo?

Esistono entrambi i tipi di TLB: quelle contenenti entry per più processi e quelle contenente solo entry per il processo attualmente attivo. Queste ultime necessitano di funzionalità di azzeramento/reset in corrispondenza a ciascun context switch



**Domanda 4**

Risposta non data

Punteggio max.:

3,00

**SE I RISULTATI SONO NUMERI, RIPORTARE PASSAGGI INTERMEDI RILEVANTI E/O FORMULE USATE****LE RISPOSTE SI/NO VANNO MOTIVATE**

Sia dato un sistema OS161. Si supponga di aver aggiunto  
a `kern/conf/conf.kern` le istruzioni

```
defoption exam
optfile project syscall/exam.c
```

e di aver creato in `kern/conf` il file EXAM, copiato dal file DUMBVM.

A) Si dica se sono sufficienti le azioni appena descritte, più l'esecuzione, in `kern/conf` di `./config EXAM`, affinché venga compilato il file opzionale `syscall/exam.c` quando si effettuano, in `kern/compile/EXAM`, i comandi

```
bmake depend
bmake
```

B) Quale, tra i file `exam.h` e `opt-exam.h`, viene generato automaticamente dal comando `./config EXAM`? Viene generato sempre oppure, oppure solo se in EXAM compare l'istruzione

```
options exam
```

C) Cosa contiene il file (quello generato automaticamente, di cui alla domanda precedente)?

D) Si supponga di inserire, nel file `main.c` l'istruzione

```
exam_init();
```

Tenendo conto che la funzione `exam_init()` viene realizzata nel file `syscall/exam.c`, è possibile fare in modo che l'istruzione sia presa in considerazione e compilata solamente nelle versioni del kernel in cui viene abilitata l'opzione `exam`, mediante le istruzioni seguenti?

```
#ifdef OPT_EXAM
    exam_init();
#endif
```

Dire anche cosa è necessario includere in un file `.c` per poter usare questo tipo di opzioni

---

A) Si dica se sono sufficienti le azioni appena descritte, più l'esecuzione, in `kern/conf` di `./config EXAM`, affinché venga compilato il file opzionale `syscall/exam.c` quando si effettuano, in `kern/compile/EXAM`, i comandi

```
bmake depend
bmake
```

NO. Non sono sufficienti. Le istruzioni definiscono l'opzione e la dipendenza del file `syscall/exam.c` dall'opzione. Ma occorre abilitare/attivare tale opzione, nel file EXAM. Diversamente il file `exam.c` non solo non verrà compilato, ma neppure considerato nella generazione delle dipendenze.

B) Quale, tra i file `exam.h` e `opt-exam.h`, viene generato automaticamente dal comando `./config EXAM`? Viene generato sempre oppure, oppure solo se in EXAM compare l'istruzione

```
options exam
```

`opt-exam.h`. Il file viene generato anche se EXAM non contiene `options exam`. Affinchè il file `.h` venga generato, è sufficiente che `conf.kern` contenga la definizione dell'opzione `exam`. Un eventuale file `exam.h` può, se necessario/opportuno, essere creato "manualmente", e va aggiunto, come opzionale oppure no, a seconda dei casi, in `conf.kern`.

C) Cosa contiene il file (quello generato automaticamente, di cui alla domanda precedente)?

`opt-exam.h`, a parte la protezione dall'inclusione multipla, contiene una sola direttiva al pre-compilatore:

```
#define OPT_EXAM 1
```

oppure

```
#define OPT_EXAM 0
```

in funzione del fatto che EXAM contenga o meno l'istruzione `options exam`

D) Si supponga di inserire, nel file `main.c` l'istruzione

```
exam_init();
```

Tenendo conto che la funzione `exam_init()` viene realizzata nel file `syscall/exam.c`, è possibile fare in modo che l'istruzione sia presa in considerazione e compilata solamente nelle versioni del kernel in cui viene abilitata l'opzione `exam`, mediante le istruzioni seguenti?

```
#ifdef OPT_EXAM
    exam_init();
#endif
```

Dire anche cosa è necessario includere in un file `.c` per poter usare questo tipo di opzioni

No. Non è possibile perché invece di `#ifdef` occorre `#if`, in quanto `OPT_EXAM` è sempre definita (vale 0 oppure 1).

Occorre utilizzare la compilazione condizionale, sfruttando la macro `OPT_EXAM`, definita in `opt-exam.h`. Quindi l'intestazione del file `main.c` (oppure uno dei `.h` da esso inclusi) dovrà contenere

```
#include "opt-exam.h"
```

e l'istruzione da condizionare (in `main.c`) andrà scritta come

```
#if OPT_EXAM  
exam_init();  
#endif
```

**Domanda 5**

Risposta non data

Punteggio max.:

3,00

**TUTTE LE RISPOSTE SÌ / NO DEVONO ESSERE MOTIVATE. QUANDO I RISULTATI SONO NUMERI, SONO NECESSARI IL RISULTATO FINALE E I RELATIVI PASSI INTERMEDI (O FORMULE)**

Si consideri un processo utente su OS161

- A) Spiegare brevemente la differenza tra switchframe e trapframe. Quale viene utilizzato per avviare un processo utente? Quale per gestire una chiamata di sistema? Quale per la thread\_fork?
- B) Perché i dispositivi di IO sono mappati su kseg1? Sarebbe possibile mappare un dispositivo di IO in kseg0?
- C) Un processo utente può eseguire una write(fd, buf, nb), dove buf è un indirizzo in kseg0?
- 

- A) Spiegare brevemente la differenza tra switchframe e trapframe. Quale viene utilizzato per avviare un processo utente? Quale per gestire una chiamata di sistema? Quale per la thread\_fork?

Entrambe le strutture vengono utilizzate per salvare il contesto del processo (registri + altre informazioni): lo switchframe viene utilizzato per il context switch (un cambio del processo in esecuzione su una cpu), il trapframe è correlato a una trap, il che significa che siamo ancora nel contesto del processo ma in modalità kernel. La funzione runprogram utilizza un trapframe per avviare un processo utente con mips\_usermode.

Una chiamata di sistema sfrutta il trapframe, poiché viene attivato come trap. La thread\_fork usa uno switchframe in quanto prepara un thread pronto per essere inserito nella coda ready.

- B) Perché i dispositivi di IO sono mappati su kseg1? Sarebbe possibile mappare un dispositivo di IO in kseg0?

I dispositivi di IO devono essere mappati nello spazio del kernel. Sono mappati in kseg1 in quanto non è memorizzato nella cache: un dispositivo di IO non può essere letto / scritto utilizzando la cache, poiché qualsiasi operazione di lettura / scrittura deve essere eseguita sul dispositivo di IO. Per lo stesso motivo, il dispositivo non può essere mappato su kseg0, che è memorizzato nella cache.

- C) Un processo utente può eseguire una write(fd, buf, nb), dove buf è un indirizzo in kseg0?

NO, per mancanza di privilegi. Il puntatore buf è l'origine dell'operazione di scrittura, mentre la destinazione è il file, che potrebbe essere un normale file o la console. In generale, qualsiasi indirizzo di memoria legale è corretto come sorgente, ma, dato un processo utente, un puntatore legale dovrebbe essere mappato nello spazio utente, quindi kseg0 è proibito.