

Domanda n1

SE I RISULTATI SONO NUMERI, RIPORTARE PASSAGGI INTERMEDI RILEVANTI E/O FORMULE USATE
LE RISPOSTE SÌ/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" del 95%. 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 200 ns, si calcoli il tempo effettivo di accesso (EAT) per il caso proposto (TLB hit ratio = 95%), assumendo che il tempo di accesso alla TLB sia trascurabile.
- B) Si consideri ora la frequenza di page fault p . Ipotizzando che un page fault sia servito in 5 ms, quanto deve valere p affinché si possa garantire un degrado massimo del 20% per EAT (causato sia della TLB che dei page fault)? Si tratta di un valore massimo o minimo per p ?
- C) Al fine di valutare un algoritmo A di sostituzione pagine, si sono simulate/provate tre stringhe di riferimento w_1, w_2, w_3 , aventi lunghezza, rispettivamente, $\text{len}(w_1)=3 \cdot 10^6$, $\text{len}(w_2)=10^6$, $\text{len}(w_3)=2 \cdot 10^6$. Le simulazioni hanno generato, rispettivamente, 120, 50 e 75 page fault. Le tre stringhe hanno probabilità (di rappresentare una generica esecuzione nel sistema reale) $p_1=0.4$, $p_2=0.2$, $p_3=0.4$. Si calcoli la probabilità empirica f (frequenza attesa) di un page fault nel sistema reale.

A) Calcolo EAT_{PT} con TLB hit ratio = 95%:

$T_{RAM} = 200 \text{ ns}$ (RAM access time)
 $h_{TLB} = 0.95$ (TLB hit ratio)
 2 level (hierarchical) PT => 2 reads for PT lookup

$$EAT_{PT} = h_{TLB} \cdot T_{RAM} + (1 - h_{TLB}) \cdot 3T_{RAM} = (1 + 2 \cdot (1 - h_{TLB})) T_{RAM} = 1.1 \cdot T_{RAM} = 220 \text{ ns}$$

B) Calcolo p per garantire un degrado massimo del 20% per EAT.
 Si tratta di un valore massimo o minimo per p ?

$T_{PF} = 5 \text{ ms}$ (PF service time)

$$\begin{aligned} EAT_{PF} &= (1-p) \cdot EAT_{PT} + p \cdot T_{PF} = (1-p) \cdot 220 + p \cdot 5 \cdot 10^6 \text{ ns} \\ EAT_{PF} &\leq 1.2 \cdot T_{RAM} = 240 \text{ ns} \\ (1-p) \cdot 220 + p \cdot 5 \cdot 10^6 &\leq 240 \\ (5 \cdot 10^6 - 220) \cdot p &\leq 240 - 220 \\ (5 \cdot 10^6) \cdot p &\leq 240 - 220 \text{ (removed negligible term)} \\ p &\leq 20/5 \cdot 10^{-6} = 4 \cdot 10^{-6} \end{aligned}$$

Valor minimo o massimo (dire perché)? Valor massimo, vista la disequazione, oppure osservando che al crescere della probabilità di Page Fault aumenta EAT_{PF} , che deve essere inferiore e un limite massimo

C) Calcolo la probabilità empirica f di un page fault nel sistema reale, date stringhe w_1, w_2, w_3 , $\text{len}(w_1)=3 \cdot 10^6$, $\text{len}(w_2)=10^6$, $\text{len}(w_3)=2 \cdot 10^6$, probabilità $p_1=0.4$, $p_2=0.2$, $p_3=0.4$ e numeri di page fault 120, 50, 75:

$F_1=120, F_2=50, F_3=75$

$$\begin{aligned} f &= p_1 \cdot F_1 / \text{len}(w_1) + p_2 \cdot F_2 / \text{len}(w_2) + p_3 \cdot F_3 / \text{len}(w_3) = \\ &= 0.4 \cdot 120 / (3 \cdot 10^6) + 0.2 \cdot 50 / 10^6 + 0.4 \cdot 75 / (2 \cdot 10^6) = \\ &= (16 + 10 + 15) \cdot 10^{-6} = 4.1 \cdot 10^{-5} \end{aligned}$$

Domanda 2

Completo.

Punteggio max.: 100,00

Domanda n2

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

Un file eseguibile contiene, in sequenza:

- un header di dimensione 5KB,
- un segmento di testo (programma e altro) di dimensione 10.5MB,
- un segmento dati (globali) di dimensione 9 MB.

Il file viene salvato in un file system F basato su inode standard (10 puntatori diretti, 1 indiretto singolo, 1 doppio, 1 triplo), basati su blocchi di dimensione 4KB e indici (puntatori a blocchi) da 32bit.

Si risponda alle domande seguenti:

- E' necessario che il segmento di testo e il segmento dati inizino all'inizio di un blocco?
- Quale è la dimensione (indipendente dal file system) del file e quale è la sua occupazione (misurata in blocchi) nel file system F?
- Calcolare la frammentazione interna (per i soli blocchi di dato)
- Quanti blocchi di indice sono necessari per rappresentare il file? (dire anche a quali livelli appartengono, se indiretto singolo, doppio e/o triplo)
- Supponendo di mantenere invariati l'header e il segmento di testo, quanto dovrebbe essere grande il segmento dati per rendere necessario il livello indiretto triplo?

A) E' necessario che il segmento di testo e il segmento dati inizino all'inizio di un blocco?

NO. Perché il contenuto del file è indipendente dal file system e dal disco in cui viene memorizzato. Il formato del file eseguibile viene trattato a un livello più alto rispetto all'implementazione del file system e all'IO su disco.

B) Quale è la dimensione (indipendente dal file system) del file e quale è la sua occupazione (misurata in blocchi) nel file system F?

$$|file| = |header| + |text segment| + |data segment| =$$

$$= 5KB + 10.5MB + 9MB = 5 + 19.5 \cdot 1024 KB = 5 + 20480 - 512 KB = 19973 KB = 19.5 MB$$

$$occupazione = \text{ceil}(19973 KB / 4KB) = 4994 \text{ blocchi} \quad (\text{ceil(): intero superiore})$$

C) Calcolare la frammentazione interna (per i soli blocchi di dato)

$$framm = 4994 \cdot 4KB - 19973KB = 19976 - 19973 = 3KB \text{ (3/4 dell'ultimo blocco)}$$

Si noti che la frammentazione interna potrebbe essere calcolata trascurando i due segmenti tex/data in quanto multipli di un blocco

D) Quanti blocchi di indice sono necessari per rappresentare il file? (dire anche a quali livelli appartengono, tre indiretto singolo, doppio e/o triplo)

Un blocco contiene fino a $4KB/4B = 1K = 1024$ indici

ND: numero blocchi dato

NI: numero blocchi indice

$$ND_0 = 10 \text{ direct data blocks}$$

$$NI_0 = 0 \text{ (no index blocks)}$$

$$ND_1 = 1K = 1024 \text{ single indirect data blocks}$$

$$NI_1 = 1 \text{ (1 index block)}$$

$$ND_2 = 4994 - (10 + 1024) = 3960 \text{ double indirect data blocks}$$

$$NI_2 = 1 \text{ outer (first level) index block} + \text{ceil}(3960/1024) = 4 \text{ inner (second level) index blocks} = 5 \text{ index blocks}$$

$$NI_{tot} = NI_1 + NI_2 = 1 + 5 = 6$$

E) Supponendo di mantenere invariati l'header e il segment di testo, quanto dovrebbe essere grande il segmento dati per rendere necessario il livello indiretto triplo?

$$|file|_{3,min} = 1B + |file|_{2,MAX} = 1B + ND_{2,MAX} \cdot 4KB =$$

$$= 1B + (10 + 1K + 1M) \cdot 4KB = 4GB + 4MB + 40KB + 1B$$

$$|data segment|_{min} = |file|_{3,min} - |header| - |text segment| =$$

$$= 4GB + 4MB + 40KB + 1B - 5B - 10.5 MB = 4GB - 6.5MB = 4089.5MB$$

Domanda n3

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

Un processo user vuole effettuare un input da un dispositivo di IO a caratteri (character device) mediante una strategia basata su polling del dispositivo. Per evitare un'attesa troppo lunga (nel loop di polling) da parte del driver del dispositivo, si chiede all'autore del programma eseguito dal processo user, di fare direttamente polling mediante un loop di lettura del registro di stato del dispositivo.

A) E' possibile effettuare questa operazione? (se sì, dire come, se no, dire perché)

B) Supponendo che il dispositivo sia associato alla tastiera e lo si gestisca in interrupt, ci si aspetta che venga generato un interrupt per ogni carattere ricevuto oppure un interrupt per ogni "invio" (cioè enter, fine-riga)? (motivare)

Dato un altro dispositivo di IO, gestito a blocchi:

C) E' possibile che una sola scrittura (mediante write (/)), tenti di fare output un vettore di dimensione più grande di un blocco?

D) Dato il driver che realizza la scrittura, supponendo che il sistema sia dotato di DMA controller, è possibile che il driver esegua la scrittura in modalità "programmed IO", senza sfruttare il DMA controller, oppure no? Perché?

A) E' possibile effettuare questa operazione? (se sì, dire come, se no, dire perché)

NO. Perché un dispositivo di IO richiede istruzioni privilegiate, con la CPU in modalità kernel. Un processo user non ha accesso al dispositivo e non ne può leggere il registro di stato, a meno che sia implementata una opportuna system call in grado di fornire tale servizio.

B) Supponendo che il dispositivo sia associato alla tastiera e lo si gestisca in interrupt, ci si aspetta che venga generato un interrupt per ogni carattere ricevuto oppure un interrupt per ogni "invio" (cioè enter, fine-riga)? (motivare)

Ci si attende un interrupt per ogni carattere. La tastiera è un dispositivo a caratteri. Una eventuale bufferizzazione dei caratteri in una riga, tale da poter effettuare correzioni prima dell'invio, va gestita via software (a meno di avere una tastiera speciale con queste caratteristiche)

Dato un altro dispositivo di IO, gestito a blocchi,

C) E' possibile che una sola scrittura (mediante write (/)), tenti di fare output un vettore di dimensione più grande di un blocco?

SI: E' possibile, né il formato di un file né la dimensione di un blocco possono vincolare l'operazione write(). Se il vettore è più grande di un blocco, verranno scritti più blocchi.

D) Dato il driver che realizza la scrittura, supponendo che il sistema sia dotato di DMA controller, è possibile che il driver esegua la scrittura in modalità "programmed IO", senza sfruttare il DMA controller, oppure no? Perché?

SI: Se è disponibile. Il fatto che esista un DMA controller non vincola il driver a utilizzarlo. Ovviamente le prestazioni risulteranno inferiori.

Domanda n4

SE I RISULTATI SONO NUMERI, RIPORTARE PASSAGGI INTERMEDI RILEVANTI E/O FORMULE USATE
LE RISPOSTE SÌ/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, come 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?

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. Affinché 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, come 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?

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
```