

System and device programming

Iniziato	venerdì, 28 maggio 2021, 17:15
Stato	Completato
Terminato	sabato, 29 maggio 2021, 17:46
Tempo impiegato	1 giorno
In ritardo	23 ore 11 min.
Valutazione	0,00 su un massimo di 14,00 (0%)

Domanda 1

Risposta non data

Punteggio max.:
3,00**WHEN RESULTS ARE NUMBERS, RELEVANT INTERMEDIATE STEPS (OR FORMULAS) ARE NEEDED****ALL YES/NO ANSWERS MUST BE EXPLAINED/MOTIVATED**

Consider a virtual memory system based on paging, with a byte addressable RAM. The system has a TLB (Translation Look-aside Buffer), with an experimentally measured hit ratio of 98%. A two level page-table is used, based on splitting a 64-bit logical address (from MSB to LSB) into 3 parts: p_1 , p_2 and d , respectively of 40 bits, 12 bits, and 12 bits. No other data structures (such as hash tables or inverted page tables) are used. Virtual memory is managed with demand paging.

Answer the following questions:

- A) Assuming the RAM memory has an access time $T_{RAM} = 300$ ns, calculate the effective access time (EAT_{PT}) for the proposed case (TLB hit ratio = 98%), assuming that the TLB lookup time is negligible.
- B) Now consider the page fault frequency p . Assuming that a page fault is served in 5 ms, what value should p take in order to guarantee a maximum performance decrease of 20% for EAT_{PF} (caused by both the TLB and the page faults) with respect to T_{RAM} ? Is this a maximum or minimum value for p ?
- C) In order to evaluate a page replacement algorithm A, three reference strings w_1 , w_2 , w_3 have been simulated / tested, having length, respectively, $\text{len}(w_1)=10^6$, $\text{len}(w_2)=2*10^6$, $\text{len}(w_3)=5*10^6$. The simulations generated 200, 100 and 50 page faults respectively. The three strings have probability (of representing a generic execution in the real system) $p_1=0.5$, $p_2=0.2$, $p_3=0.3$. Compute the empirical probability f (expected frequency) of a page fault in the real system.

A) EAT_{PT} calculation with TLB hit ratio = 98%:

$T_{RAM} = 300$ ns (RAM access time)

$h_{TLB} = 0.98$ (TLB hit ratio)

2 level (hierarchical) PT => 2 reads for PT lookup

$$EAT_{PT} = h_{TLB} * T_{RAM} + (1-h_{TLB}) * 3T_{RAM} = (1 + 2*(1-h_{TLB}))T_{RAM} = 1.04 * T_{RAM} = 312 \text{ ns}$$

B) Calculation of P to guarantee a maximum degradation of 20% for EAT_{PF} . Is this a maximum or minimum value for p ?

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

$$EAT_{PF} = (1-p) \cdot EAT_{PT} + p \cdot T_{PF} = (1-p) \cdot 312 + p \cdot 5 \cdot 10^6 \text{ ns}$$

$$EAT_{PF} \leq 1.2 \cdot T_{RAM_b} = 360 \text{ ns}$$

$$(1-p) \cdot 312 + p \cdot 5 \cdot 10^6 \leq 360$$

$$(5 \cdot 10^6 - 312) \cdot p \leq 360 - 312$$

$$(5 \cdot 10^6) \cdot p \leq 360 - 312 \text{ (removed negligible term)}$$

$$p \leq 48/5 \cdot 10^{-6} = 9.8 \cdot 10^{-6} \approx 10^{-5}$$

Minimum or maximum value (explain why)?

Maximum value, given the inequality; or by observing that as the probability of Page Fault increases, EAT_{PF} increases, which must be less than a maximum limit

C) Calculation of the empirical probability f of a page fault in the real system, given the strings w_1, w_2, w_3 , $\text{len}(w_1)=10^6$, $\text{len}(w_2)=2 \cdot 10^6$, $\text{len}(w_3)=5 \cdot 10^6$, probability $p_1=0.5$, $p_2=0.2$, $p_3=0.3$ and page fault numbers 200, 100, 50:

$$F_1=200, F_2=100, F_3=50$$

$$\begin{aligned} f &= p_1 \cdot F_1 / \text{len}(w_1) + p_2 \cdot F_2 / \text{len}(w_2) + p_3 \cdot F_3 / \text{len}(w_3) = \\ &= 0.5 \cdot 200 / 10^6 + 0.2 \cdot 100 / (2 \cdot 10^6) + 0.3 \cdot 50 / (5 \cdot 10^6) = \\ &= (100 + 10 + 3) \cdot 10^{-6} = 1.13 \cdot 10^{-4} \end{aligned}$$

Domanda 2

Risposta non data

Punteggio max.:
3,00**WHEN RESULTS ARE NUMBERS, RELEVANT INTERMEDIATE STEPS (OR FORMULAS) ARE NEEDED****ALL YES/NO ANSWERS MUST BE EXPLAINED/MOTIVATED**

An executable file contains three sections:

- a header of 3KB size,
- a text segment (program and other) of size 11MB,
- a (global) data segment of size 9.5 MB.

The file is saved in a file system F based on standard iNodes (10 direct pointers, 1 single indirect, 1 double, 1 triple), with 2KB size blocks and 32bit indexes (block pointers).

Answer the following questions:

- A) Is it necessary for the text segment and the data segment to begin at a block boundary (address multiple of the block size)?
- B) What is the size (regardless of the file system) of the file and what is its occupation (measured in blocks) in the file system F?
- C) Compute the internal fragmentation (for data blocks only).
- D) How many index blocks are needed to represent the file? (Also say which level they belong to, if single indirect, double and/or triple)
- E) Assuming the header and the text segment remain unchanged, what could the maximum size of the data segment be to avoid using the triple indirect level ?

A) Is it necessary for the text segment and the data segment to begin at a block boundary?

NO. Because the content of the file is independent from the file system and the disk on which it is stored. The format of the executable file is processed at a higher level than the implementation of the file system and the IO on disk.

B) What is the size (regardless of the file system) of the file and what is its occupation (measured in blocks) in the file system F?

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

$$= 3KB + 11MB + 9.5MB = 3 + 20.5 \cdot 1024\ KB = 5 + 20480 + 512\ KB = 20995\ KB = 20.5\ MB$$

$$usage = \text{ceil}(20995\ KB / 2KB) = 10498\ blocks \quad (\text{ceil}(): \text{ceiling operator})$$

C) Compute internal fragmentation (for data blocks only).

fragmentation = $10498 * 2KB - 20995KB = 20996 - 20995 = 1KB$ (1/2 of the last block)

D) How many index blocks are needed to represent the file? (Also say which levels they belong to, if single indirect, double and/or triple)

A block contains up to $2KB/4B = 0.5K = 512$ indexes

ND: number of data blocks

NI: number of index blocks

$ND_0 = 10$ direct data blocks

$NI_0 = 0$ (no index blocks)

$ND_1 = 1K = 512$ single indirect data blocks

$NI_1 = 1$ (1 index block)

$ND_2 = 10498 - (10 + 512) = 9976$ double indirect data blocks

$NI_2 = 1$ outer (first level) index block + $\text{ceil}(9976/512) = 20$ inner (second level)

index blocks = 21 index blocks

$NI_{\text{tot}} = NI_1 + NI_2 = 1 + 21 = 22$

E) Assuming the header and the text segment remain unchanged, what could the maximum size of the data segment be to avoid using the triple indirect level ?

$|file|_{2, \text{MAX}} = ND_{2, \text{MAX}} * 2KB =$

$= (10 + 0.5K + 0.25M) * 2KB = 0.5GB + 1MB + 20KB$

$|data\ segment|_{\text{MAX}} = |file|_{2, \text{MAX}} - |header| - |text\ segment| =$

$= 0.5GB + 1MB + 20KB - 3B - 11\ MB \approx 0.5GB - 10MB = 502MB$

Domanda 3

Risposta non data

Punteggio max.:
2,00**WHEN RESULTS ARE NUMBERS, RELEVANT INTERMEDIATE STEPS (OR FORMULAS) ARE NEEDED****ALL YES/NO ANSWERS MUST BE EXPLAINED/MOTIVATED**

A user process wants to make an input from a character IO device using a polling strategy. To avoid a too long wait (in the polling loop) by the device driver, the author of the program executed by the user process is asked to directly poll the device using a reading loop, in the user program, of the device status register.

A) Is it possible to carry out this operation? (if yes, say how, if not, say why)

B) Assuming that the device is associated to the keyboard and that it is managed in interrupt, do you expect that an interrupt will be generated for each character received or an interrupt for each "return" (i.e. enter / end of line)? (explain why)

Given another IO block device:

C) Is it possible that a single write (through write()), tries to output an array larger than a block?

D) Given the driver who performs the writing, can this operation be carried out without the DMA controller (so by the CPU), when the system is equipped with a DMA controller, or not? Why?

A) Is it possible to carry out this operation? (if yes, say how, if not, say why)

NO. Because an IO device requires privileged instructions, with the CPU in kernel mode. A user process does not have access to the device and cannot read its status register, unless a suitable system call capable of providing this service is implemented.

B) Assuming that the device is associated with the keyboard and that it is managed in interrupt, is it expected that an interrupt will be generated for each character received or an interrupt for each "return" (i.e. enter / end of line)? (explain why)

An interrupt is expected for each character. The keyboard is a character device. Any buffering of the characters in a row, in order to be able to make corrections before sending, must be managed via software (unless you have a special keyboard with these characteristics)

Given another block-managed IO device:

C) Is it possible that a single write (through write()), tries to output an array larger than a

block?

YES: It is possible, neither the format of a file nor the size of a block can constrain the write() operation. If the array is larger than a block, multiple blocks will be written.

D) Given the driver who performs the writing, can this operation be carried out without the DMA controller (so by the CPU), when the system is equipped with a DMA controller, or not? Why?

YES, of course: Though the DMA controller would obtain a better performance, nothing prevents a driver to perform the operation avoiding the DMA controller

Domanda 4

Risposta non data

Punteggio max.:
3,00

WHEN RESULTS ARE NUMBERS, RELEVANT INTERMEDIATE STEPS (OR FORMULAS) ARE NEEDED
ALL YES/NO ANSWERS MUST BE EXPLAINED/MOTIVATED

An OS161 system is given. Suppose you have added the following instructions to `kern/conf/conf.kern`

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

and you have created the `PROJECT` file in `kern/conf`, copied from the `DUMBVM` file.

A) Tell if the actions described above, plus the execution in `kern/conf` of `./config PROJECT`, are sufficient for the optional file `syscall/project.c` to be compiled when you execute, in `kern/compile/PROJECT`, the commands

```
bmake depend
bmake
```

B) Which one between the `project.h` and `opt-project.h` files is automatically generated by the command `./config PROJECT`? It is always generated, or only if the following instruction appears in `PROJECT`?

```
options project
```

C) What does the file contain (the automatically generated one, referred by the previous question)?

D) Suppose you insert in `main.c` the instruction

```
project_init();
```

Taking into account that the `project_init()` function is implemented in the `syscall/project.c` file, how can the instruction be taken into consideration and compiled only in the kernel versions in which the `project` option is enabled?

A) Tell if the actions described above, plus the execution in `kern/conf` of `./config PROJECT`, are sufficient for the optional file `syscall/project.c` to be compiled when you carry out, in `kern/compile/PROJECT`, the commands

```
bmake depend
bmake
```

NO. They are not enough. The instructions define the option and the dependency of the file `syscall/project.c` on the option. But you need to enable this option, in the `PROJECT` file. Otherwise the `project.c` file will be not only not compiled, but not even considered in the generation of dependencies.

B) Which of the `project.h` and `opt-project.h` files is automatically generated by the command `./config PROJECT`? It is always generated or, or only if in `PROJECT` the following instruction appears?

```
options project
```

`opt-project.h`. The file is generated even if `PROJECT` does not contain `options project`. In order for the `.h` file to be generated, it is sufficient for `conf.kern` to contain the definition of the `project` option. A possible `project.h` file can, if necessary/appropriate, be created "manually", and must be added, as optional or not, as appropriate, in `conf.kern`.

C) What does the file contain (the automatically generated one, referred by the previous question)?

`opt-project.h`, apart from multiple inclusion protection, contains only one directive to the pre-compiler:

```
#define OPT_PROJECT 1
```

or

```
#define OPT_PROJECT 0
```

depending on whether or not `PROJECT` contains the `options project` statement

D) Suppose you insert in `main.c` the instruction

```
project_init();
```

Taking into account that the `project_init()` function is implemented in the `syscall/project.c` file, how can the instruction be taken into consideration and compiled only in the kernel versions in which the `project` option is enabled?

Conditional compilation must be used, using the `OPT_PROJECT` macro, defined in `opt-project.h`. So the header of the `main.c` file (or one of the `.h` included by it) must contain

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

and the instruction to be conditioned (in `main.c`) will be written as

```
#if OPT_PROJECT
project_init();
#endif
```

Domanda 5

Risposta non data

Punteggio max.:
3,00

WHEN RESULTS ARE NUMBERS, RELEVANT INTERMEDIATE STEPS (OR FORMULAS) ARE NEEDED
ALL YES/NO ANSWERS MUST BE EXPLAINED/MOTIVATED

An OS161 system is given, in the basic version. Consider the definitions of `struct thread` and `struct proc`, partially reported here:

```
struct thread {
    char *t_name;
    ...
    void *t_stack;
    struct switchframe *t_context;
    struct cpu *t_cpu;
    struct proc *t_proc;
    /* add more here as needed */
};

struct proc {
    ...
    unsigned p_numthreads;
    struct addrspace *p_addrspace;
    struct vnode *p_cwd;
    /* add more here as needed */
};
```

Answer the following questions:

A) Say how the two `struct` are related/connected in case a process has more than one thread.

B) Say if, given a pointer `p` to a `struct thread`, it is possible to get the list of all other threads of the same process and print their names? If it is not possible, propose how to modify (add fields) to `struct proc` and/or `struct thread`, in order to carry out the operation (given a thread, list other threads of the same process). Write a kernel function `printOtherThreadNames(struct thread *t)` that does the work

C) Consider the `struct proc`. Explain why, although the `p_addrspace` field allows you to reach the stack pointer of a user process, also the `struct thread` contains a stack pointer. Is this a redundant information (there is only one stack to which the thread and indirectly the process point), or are they two different stacks?

D) Briefly explain the role of the `t_context` pointer in the `struct thread`.

A) Say how the two `struct` are related/connected in case a process has more than one

thread.

The `struct thread` points (`t_proc` field) to the `struct proc` of the process. The `t_proc` does not contain pointers to the threads of the process, it only contains the count of these threads (`p_numthreads` field).

B) Say if, given a pointer `p` to a `struct thread`, it is possible to get the list of all other threads of the same process and print their names? If it is not possible, propose how to modify (add fields) to `struct proc` and/or `struct thread`, in order to carry out the operation (given a thread, list other threads of the same process). Write a kernel function `printOtherThreadNames(struct thread *t)` that does the work

In the current version, the operation is not possible because the `struct proc` does not point to an array or to a list of pointers to the relevant threads. It is excluded, because considered expensive, the possibility of iterating over all the threads active in the kernel (as far as you can do it) by testing for each one if it points to the given process.

One way to allow this could be to add a pointer to the list of its threads to the `struct proc`,

```
struct proc {
    ...
    /* add more here as needed */

    struct thread *p_threadListHead;
}
```

adding a field to the `struct thread` that acts as a link/next for that list

```
struct thread {
    ...
    /* add more here as needed */
    struct thread *t_threadListNext;
}

printProcThreadNames(struct proc *p) {
    struct thread *t;
    /* no error check: assume t_name is not NULL */
    for (t=p->p_threadListHead; t!=NULL; t=t->t_threadListNext)
        kprintf(t->t_name);
}
```

C) Consider the `struct proc`. Explain why, although the `p_addrspace` field allows you to reach the stack pointer of a user process, also the `struct thread` contains a stack pointer. Is this a redundant information (there is only one stack to which the thread and indirectly the process point), or are they two different stacks?

These are two different stacks: the one pointed by the `addrspace` is the user stack of the process (currently created within `load_elf()`), the one pointed by the `t_stack` field of the `struct thread` is the kernel stack, used, for example, to save the switchframe and the trapframe.

D) Briefly explain the role of the `t_context` pointer in the `struct thread`.

It points to the thread's switchframe, which allows to save/restore the context of the thread at each context switch (change of the thread running on the CPU). For example, see the `thread_switch()` and `switchframe_switch()` functions.