CSCI 3431.1 Fall, 2017 – Assignment 4 Operating Systems

0.1 Practice Exercises

1. [6 points] Consider the Intel address-translation scheme shown in Figure 8.22 in your textbook.

a. Describe all the steps taken by the Intel Pentium in translating a logical address into a physical address.

Solutions:

After compiling the source code to binary runnable application, the binary application use the logical address in order to operate the memory.

In order to access the physical memory, the logical address of each application should be translate the linear address. And then, the linear address will be translated into physical address through paging unit in Intel Pentium architecture.

We can simply think that the data structure of segment descriptor includes three parts: base address, limit of this segment and some extra attributes of this segment. In order to store the whole segment descriptors, a segment descriptor table is created in Intel Pentium architecture.

There are more than one descriptor table, for example, global descriptor table (GDT) is charge of the share memory for different process. Each task's private request will be put in LDT (local descriptor table).

There are two parts of one logic address: selector (segment descriptor) and offset.

Firstly, the Intel Pentium CPU will find the base from descriptor table according to the segment descriptor.

Secondly, if the request memory size is less than limit, it's a valid allocation request, the base address of segment will be added with the offset. The real 32-bit linear address will be gotten.

If there is no paging unit, this linear address is the exact physical address.

Thirdly, the top 10 bits in the linear address points out the first level page directory table. The middle 10 bits in the linear address points out the second level page table. Getting the physical address base from the second level table, add the lower 12 bits of the linear address, finally get the physical memory address.

b. What are the advantages to the operating system of hardware that provides such complicated memory translation?

Solutions:

According to the application development, during the compiling stage, the program will be split into different segments, each segment uses its own logic address. For example, static variables, new objects allocated during program running stage (in heap), local variables in different functions (in stack).

The first advantage is the segment supports the compiler to create different segments for different parts in one programme.

The second advantage is different segments can share the data with more convenient.

The third is it’s possible to protect the OS process in this type of impalement. For example, some segment is only of the OS process, this design will be possible with this memory translation technology.

The last one is the frame maybe not continuous, but the logic linear address must be continuous. So the OS can schedule different memory frame for different process with different algorithms.

c. Are there any disadvantages to this address-translation system? If so, what are they? If not, why is this scheme not used by every manufacturer?

Solutions:

The first disadvantage is that the compiler needs to be modified and improved in order to use different segments. The cost will be increased.

The second disadvantage is about memory utilization. For example, the GDT and paging information must be put in memory, the memory will be more than those implement without segment and paging.

The last one is about errors. Due to some technical bugs, the OS maybe calculate a wrong physical address through the complicate translation stage. This means that this technology requires the Operation System more features to support it. It the OS implement is not efficient, the performance of the whole system will decline.

2. [4 points] can you think of any situations where supporting virtual memory would be a bad idea, and what would be gained by not having to support virtual memory? Explain

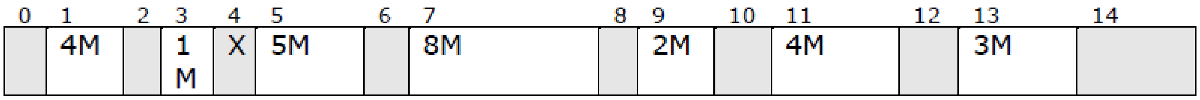
Solutions:

If there is not virtual memory feature for one operation system, the software developer and programmer must consider to manage the memory very carefully during the software development because if one program could not get sufficient memory allocation, the program will stop crudely.

Some program need a couple of memory for swap the temporary data in order to calculate some mathematic problems, for example, the matrix operations. In this case, if the operation system support virtual memory, the programmer maybe think that the memory is sufficient, so put a couple of variables into the memory at the same time. But in fact, the OS use hard disk to simulate the memory, the performance of application software will decline significantly.

From the view of system administration or operation persons’ view, if the OS do not support the virtual memory management, the software providers have to think about how to use memory more efficiently. As a result, the stability of application software will be improved.

3. [4 points] This diagram shows an example of memory configuration under dynamic partitioning, after a number of placement and swapping-out operations have been carried out. Addresses go from left to right; gray areas indicate blocks occupied by processes; white areas indicate free memory blocks. The last process placed is 2 Mbytes and is marked with an X. Only one process was swapped out after that. A new 3-Mbyte allocation request must be satisfied next. Indicate the intervals of memory where a partition will be created for the new process under the following four placement algorithms:



a. Best-fit

The number 13 block is the best option for new request because number 13 is 3MB. So the new request will be assigned to number 13 block.

b. First-fit

The first available memory block is number 1, because 4M > 3. So in Best-fit algorithms, the request for 3 MB allocation will be assigned in number 1 block.

c. Next-fit

The last memory allocation occurs in the block 4 with 2MB, so the new request with 3MB will be assigned with number 5 block because 5MB> 3MB.

d. Worst-fit

The current largest available memory block is number 7 with 8MB, so the new request with 3MB will be assigned with number 7.

4. [5 points] Consider the following segment table:

Segment	**Base**	**Length**
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses? Indicate if either addresses is illegal.

a. 0,430

430<600, it’s legal access, physical address will be 219+430=649

b. 1,10

10<14, it's legal access, physical address will be 2300+10=2310

c. 2,500

500>100, this is an illegal access.

d. 3,400

400<580, it's legal access, physical address will be 1327+580=1907

e. 4,112

112>96, this is an illegal access.

5. [6 points] We've discussed swapping as a technique used by systems to multi-task.

Although mobile systems accommodate multi-tasking, they do not typically support swapping in any form.

a. Find out why mobile systems do not support swapping.

Solutions:

Firstly, the speed of build-in hard disks in most mobile devices is very slow, it means if the operation system support the swapping feature, when the data is swapped into hard disk, the performance will decline significantly.

Secondly, the multi-tasking is a very popular feature in mobile environment. For example, the end user will switch different APPs quickly and frequently.

Thirdly, most of the mobile operation system has a notification management sub-system. For example, if a message comes in, the phone will open the Message APPs immediately. It is possible to resume one APP from memory, but from hard disk is not acceptable.

Finally, the user-friendly is the most important feature in mobile devices. The manufacturers prefer to provide a quick reaction to end user rather than letting them to wait for a long time.

b. Choose one of the available mobile OS (either Apple's iOS or Android) and discuss the strategies they implement to allow multi-tasking.

Solutions:

Getting some idea from https://android-developers.googleblog.com/2010/04/multitasking-android-way.html.

In Android, there are several thoughts when design how to support multi-tasking:

1. One android device may not support swap space due to some cost relevant thinking. For example, in those low-price devices, the manufacturers prefer to remove the swap space support in hardware level. This fact asks the Android devices must suppose there is no space swapping available during the OS is running.
2. The end user prefer to switch APPs frequently. When they listen music, if there is a new notification from snapchat, they will switch to it immediately. So the user experience for switching APPs makes a very important features.
3. Considering the limitation of hard disk's access speed, all of the APPs live in the memory, and all of the processes could not be stopped, just suspended.

So, the Android operation system select to implement the above strategies to support multi-tasking.

c. Knowing this restriction, how does this affect your next mobile application development?

Solutions:

After learning how to implement multi-tasking supports in Android, we can consider how to develop an APP about memory management.

1. The developer could not consider how to stop the process and the operation system will suspend it when the APP is not in front level (not using by the end user).
2. But the developer should release memory as soon as possible because there is a risk to be killed by OS when the memory is not sufficient. This is to say, when the OS detect that the memory is not available enough, the OS will kill some process according to the algorithms.

Self-evaluation Please answer the following questions:

[1 points] Were you able to complete this assignment? What grade are you expecting? Please justify.

Yes, I did this assignment by myself on the time, and I feel my solution has reached the expectation from the teacher.

2. **[2 points]** Describe 2-3 challenges you faced while completing this assignment. How did you tackle those challenges?

The first is try to understand how Intel Pentium to implement to translate the logical address to physical address, especially the segment and paging unit. I answered this question by reading the textbook.

The second is about the implementation of banker algorithm. Firstly, I read the textbook to understand how to judge if the current status is in a safe mode. Secondly, I convert this idea into java source codes.

3. **[2 points]** Provide a break down for the activities/milestones for this assignment. Give an estimate of hours spent on each activity. Try to be honest

Date	Activities	Hours	Outcome
23/11	Part I: Practice Exercises 1	2	Solutions
24/11	Part I: Practice Exercises 2 3 4	2	Solutions
25/11	Part I: Practice Exercises 5	3	Solutions
26/11	Part II: OPTION B - Programming Exercise: learn and design the banker algorithm	2	The skeleton of source code
27/11	Part II: OPTION B - Programming Exercise: java implementation: bankImp.java	2	Source code

Lingda Cai A00372181