CPSC 457_Muzhou,Zhai_(10106810)_T02

Assignment 1

Question 1 (written question)

Assume that a CPU pipeline has three stages, as shown in class, and each stage is handled by a separate unit, namely, fetch unit, decoding unit, and execute unit. For every instruction, the fetch unit takes 10 nsec, the decoding unit takes 0.5 nsec, and the execute unit takes 1 nsec. How many instructions per second can this CPU execute on average? Is it possible to improve the performance of this CPU? If so, how? If not, why?

From the lecture, we have learned that while executing instruction n, the CPU could be decoding instruction n+1 and fetching instruction n+2. The first instruction will take 10 +0.5 +1 = 11.5 ns. And the second instruction will take only the time of fetching and decoding, that is 10+0.5 = 10.5 ns. However, after the first and second institution, the execute time will depend on the most time consuming stage ----- the fetch unit. So all these instructions will take 10 ns. That means the average executing time for the whole pipeline is 11.5 + 10.5 + 10n / n +2. (n is the number of instructions and n >= 3) When n is tending to infinity, the average executing time for one instruction will be 10.0004 ns, approximately 10 ns. So that the instructions per can be execute on average per second is : 1s/ 10 * 10.0004 ^ -9 s = 1 * 10 ^ 8.

-Yes, it is possible for this CPU to improve the performance by adding some pipeline stages. For example, the most time consuming stage is the fetch unit, If we divide this stage into two or more sub-stages, the processing time for these sub-stages will be less than 10 ns, and the instructions of the same stages can be processed together, so that the totally executing time will be reduced.

Question 2 (written question)

What are the benefits of using a virtual machine from the operating system's perspective and from the user's perspective?

From the operating system's perspective:

-**Security**: If an insecure file runned on the virtual machine, the original computer and OS system will not be damaged.

-**System independency**: the virtual machine and the original OS system are independent from each other. For example, operations from the virtual machine very less likely needs to be disrupted from system development

-**High Availability**: The data load of the original OS system can be distributed across the virtualized machines. The virtual machine is able to ensure high availability of applications and data, as well as reduce the data loss even if the server fail.

From the user's perspective:

-**Familiar OS Interfaces**: The environments of virtual machine is built to mimic the physical one, so the user can use their familiar OS system, and have not to learn to use a new system. Users can receive similar experience and performance on a virtual network that you would with physical servers.

-**Money saving**: Since multiple different OSes can be running on the same computer by using the virtual machine, so that users can save some money.

-**System consolidation**: Providing users a clear view, and it is easier for users to manage the system.

Question 3 (written question)

Define interrupts and traps and then describe the differences. Explain why they are handled in kernel mode instead of user mode.

-Interrupts: A signal to the processor send by hardware or software, that indicate an event needs attention. Kernel talks to the device driver, requesting an operation, then the device driver response and send a interrupts. Hardware interrupts are used by hardware devices to communicate with OS for attention, while software interrupt is caused by exceptional condition or a special instruction which causes an interrupt.

-Traps: A special instruction that paused the application and executes a kernel routine configured by the OS. Usually it switches from kernel mode and invokes a predefined function.

-Difference: A trap instruction switches from user mode to kernel mode. An interrupt is used to save the state of a process during a context switch. An interrupt alerts the processor to a high-priority condition requiring the interruption of the current code the processor is executing(external event). A trap can be used to call operating system routines or to catch arithmetic errors in OS(internal event). Interrupts asynchronous with the current activity of the CPU, while trap synchronous with the current activity of the CPU. For the interrupts, the time of the event is not known and it is not predictable. For traps, it occurs during the execution of a machine instruction.

-Why they are handled in kernel mode instead of user mode: The kernel handles all interrupts as well as traps, because the kernel has the required privilege and state in most situations. In the kernel mode, all instructions, I/O, and memory are allowed and can be accessed. If they are handled by the user mode that means the processors cannot talk to hardware. A trap cannot call operating system routines or to catch arithmetic errors in OS. The processor cannot send interrupts from hardware or software that indicate an event needs attention. All these things cannot work successfully under the user mode, so that is the reason why they are handled in kernel mode instead of user mode.

Question 4 (written question)

-What are the outputs of the time commands? Copy/paste this from the shell?

```
[mzhai@zone17-eb Downloads]$ time ./readFile sample.txt
Sonnet 18
by William Shakespeare

Shall I compare thee to a summer's day?
Thou art more lovely and more temperate:
Rough winds do shake the darling buds of May,
And summer's lease hath all too short a date:
Sometime too hot the eye of heaven shines,
And often is his gold complexion dimm'd;
And every fair from fair sometime declines,
By chance, or nature's changing course, untrimm'd;
But thy eternal summer shall not fade,
Nor lose possession of that fair thou owest;
Nor shall Death brag thou wander'st in his shade,
When in eternal lines to time thou growest;
So long as men can breathe, or eyes can see,

real    0m0.003s
user    0m0.002s
sys    0m0.001s
[mzhai@zone17-eb Downloads]$ time cat sample.txt
Sonnet 18
by William Shakespeare

Shall I compare thee to a summer's day?
Thou art more lovely and more temperate:
Rough winds do shake the darling buds of May,
And summer's lease hath all too short a date:
Sometime too hot the eye of heaven shines,
And often is his gold complexion dimm'd;
And every fair from fair sometime declines,
By chance, or nature's changing course, untrimm'd;
But thy eternal summer shall not fade,
Nor lose possession of that fair thou owest;
Nor shall Death brag thou wander'st in his shade,
When in eternal lines to time thou growest;
So long as men can breathe, or eyes can see,
So long lives this, and this gives life to thee.
real    0m0.002s
user    0m0.000s
```

sys 0m0.001s

-How much time did the C++ program and cat spend in the kernel mode and user mode, respectively?

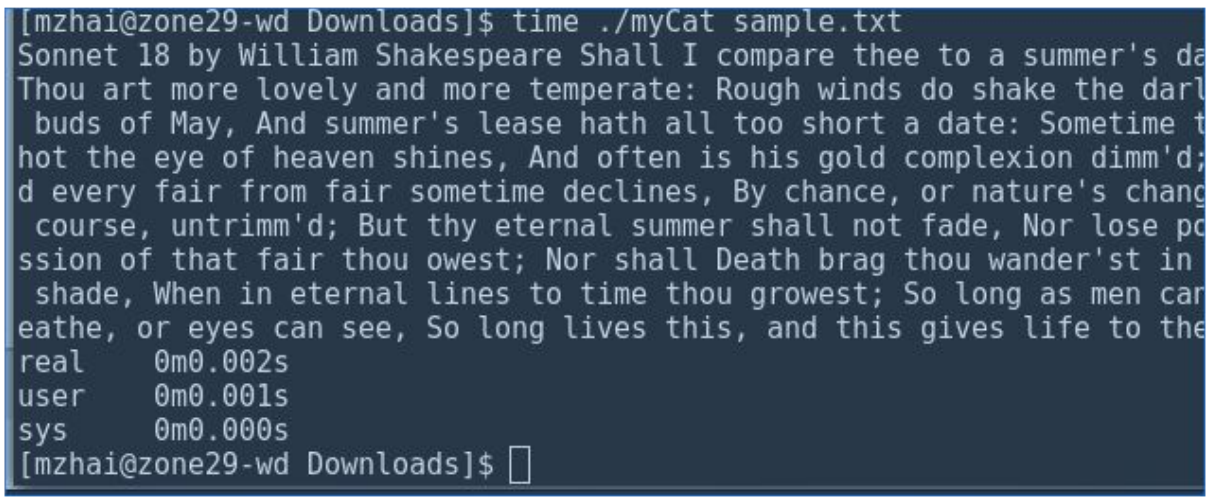
The C++ program: Kernel mode: 0.001s; User mode: 0.002s
The cat: Kernel mode: 0.001s; User mode: 0.000s

-Why is the cat program faster than the C++ program?

-Firstly, The C++ program is basically recognize and print the valid lines, that is a line by line work with using the input and output stream “fstream”. The program is based on both kernel and applications, basically much more work in the user mode, so that the time spend on the user mode is longer than in the kernel mode. However, the cat operation’s principle is concatenate files and print on the standard output. There is merely operations related to the application, so no time spend on the user mode. That is the reason compared to the c++ program, the cat operation takes less time.

-Secondly, Cat is a command that can be separate executed, it is not a shell. For example the fork/exec calls may work in the program and they works very fast. However the buffers in the readFile.cpp may works slow down due to some object construction and deletion in the background.

Question 5 (Programing Questiion)
Report
Here is the time running result from myCat.cpp file.
For this file I have modified a few lines from the original readFile.cpp file. Instead of using getLine() method reading line by line, I output the txt file as string. So that the entire running time and calls has been eliminated.



```
[mzhai@zone29-wd Downloads]$ strace -c ./myCat sample.txt
Sonnet 18 by William Shakespeare Shall I compare thee to a summer's day?
Thou art more lovely and more temperate: Rough winds do shake the darling
 buds of May, And summer's lease hath all too short a date: Sometime too
hot the eye of heaven shines, And often is his gold complexion dimm'd; An
d every fair from fair sometime declines, By chance, or nature's changing
 course, untrimm'd; But thy eternal summer shall not fade, Nor lose posse
ssion of that fair thou owest; Nor shall Death brag thou wander'st in his
 shade, When in eternal lines to time thou growest; So long as men can br
eathe, or eyes can see, So long lives this, and this gives life to thee.
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 22.33    0.000023           2        10           mprotect
 16.50    0.000017           3         6           read
 16.50    0.000017           1        29        23 open
  9.71    0.000010           2         6           fstat
  8.74    0.000009           1        16           mmap
  7.77    0.000008           8         1           munmap
  6.80    0.000007           7         1           write
  4.85    0.000005           1         6           close
  4.85    0.000005           2         3           brk
  1.94    0.000002           2         1           arch_prctl
  0.00    0.000000           0         4         3 stat
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
------ ----------- ----------- --------- --------- ----------------
100.00    0.000103                    85        27 total
[mzhai@zone29-wd Downloads]$
```

Here is the running time for the readFile.cpp after using the strace command.

```
[mzhai@zone29-wd Downloads]$ strace -c ./readFile sample.txt
Sonnet 18
by William Shakespeare

Shall I compare thee to a summer's day?
Thou art more lovely and more temperate:
Rough winds do shake the darling buds of May,
And summer's lease hath all too short a date:
Sometime too hot the eye of heaven shines,
And often is his gold complexion dimm'd;
And every fair from fair sometime declines,
By chance, or nature's changing course, untrimm'd;
But thy eternal summer shall not fade,
Nor lose possession of that fair thou owest;
Nor shall Death brag thou wander'st in his shade,
When in eternal lines to time thou growest;
So long as men can breathe, or eyes can see,
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 29.58    0.000176           6        29        23 open
 24.54    0.000146           9        16           mmap
 19.33    0.000115          12        10           mprotect
  7.06    0.000042           7         6           read
  6.22    0.000037           6         6           close
  4.54    0.000027           2        16           write
  4.03    0.000024           4         6           fstat
  2.35    0.000014          14         1           munmap
  1.51    0.000009           3         3           brk
  0.84    0.000005           5         1           arch_prctl
  0.00    0.000000           0         4         3 stat
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
------ ----------- ----------- --------- --------- ----------------
100.00    0.000595                   100        27 total
[mzhai@zone29-wd Downloads]$
```

And here is the running time for the cat command after using the strace command.

```
[mzhai@zone29-wd Downloads]$ strace -c cat sample.txt
Sonnet 18
by William Shakespeare

Shall I compare thee to a summer's day?
Thou art more lovely and more temperate:
Rough winds do shake the darling buds of May,
And summer's lease hath all too short a date:
Sometime too hot the eye of heaven shines,
And often is his gold complexion dimm'd;
And every fair from fair sometime declines,
By chance, or nature's changing course, untrimm'd;
But thy eternal summer shall not fade,
Nor lose possession of that fair thou owest;
Nor shall Death brag thou wander'st in his shade,
When in eternal lines to time thou growest;
So long as men can breathe, or eyes can see,
So long lives this, and this gives life to thee.% time     seconds  usecs/call
   calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 30.25    0.000095           8        12         8 open
 16.88    0.000053           5        10           mmap
  9.55    0.000030          30         1           write
  9.24    0.000029           7         4           mprotect
  7.96    0.000025          13         2           munmap
  7.01    0.000022           6         4         3 stat
  4.14    0.000013           3         5           fstat
  3.82    0.000012           2         6           close
  3.50    0.000011           4         3           read
  2.87    0.000009           2         4           brk
  2.23    0.000007           7         1         1 access
  1.27    0.000004           4         1           execve
  0.96    0.000003           3         1           arch_prctl
  0.32    0.000001           1         1           fadvise64
------ ----------- ----------- --------- --------- ----------------
100.00    0.000314                    55        12 total
```

Compare to the results of readFile.cpp and the file using cat command, myCat.cpp is now much faster than the readFile.cpp by the elimination of the calls and the progresses.