Exercises

**1 Please explain the following terms: (1) Response time (2) Execution time (3) Throughput (4) Benchmark**

**Sol:**

(1) Response time: How long does it take for the first response to be generated when the required work is submitted.

(2) Execution time: The time it takes for a job to run from start to finish.

(3) Generate: The maximum number of times a specific task can be completed per unit of time.

(4) Evaluation program: A set of hybrid programs that cover various calculations and are used to evaluate system performance is called a set of evaluation programs. The single program that makes up the review program group is called the evaluation program.

**2 Every three years, the number of transistors in the computer increases by 4 times, and the performance (speed) of the CPU is directly proportional to the internal transistors and the maximum clock frequency. In 1996, the x586/200 CPU contained 3 million transistors with a clock frequency of 200MHz. If an x1486/800 CPU with an 800MHz clock frequency is released in 2002, how many times faster will it be than x586/200?**

**Sol:**

times

**3 Try to explain the relationship between the execution speed of a computer and the amount of generation. What are the similarities and differences between the two?**

**Sol:**

(1) The relationship between execution speed and generation volume:

The generation volume can find the average time to complete each job.

(2) The difference between the two:

The average time does not necessarily equal the actual execution time of each job, as the computer itself may have improved the structure of the execution of instructions. For example, using pipeline processing techniques can shorten the total time required to complete multiple tasks. But for a single job, its progress time has not been shortened.

**4 It proves that the three factors of software, hardware structure, and hardware technology together affect computer performance.**

**Sol:**

|  |  |
| --- | --- |
| Hardware or software section | Impact on performance |
| Algorithm (software structure). | Affects the number of commands that the program must execute and how much I/O is generated |
| Programming languages, compilers, and computer instruction set structures (software structures). | Decide how many machine instructions will be generated |
| Machine code instructions are mapped to the circuitry (hardware structure) that implements the instruction | Determines the number of clock cycles required to execute the command. Typically, simple circuits require much more clock cycles than complex hardware circuits. |
| Processor and memory system (hardware technology). | Determine the time it takes to execute each command, as well as the clock frequency |

**5 Please try to explain how to compare the performance of different computers to be fair? Tell me your reasons.**

**Sol:**

(1) It is fairest to run the same program and compare CPU execution times.

(2) If the same program is executed, the execution time obtained will be fair. Then, according to the inverse relationship between performance and execution time, a relationship between execution time and performance can be given:



Suppose we want to evaluate the performance of computers A and B, then *Performance A* give *Performance B* They represent the work efficiency of System A and System B when running the same program and processing the same data. With such a calculation standard, we can use data fairly to represent the performance relationship between the two systems.

**6 Try to list the basic performance calculation formula for CPU calculation time and explain the meaning of each related variable.**

**Sol:**

CPU time = clock cycle time × CPI × IC

* Clock cycle time: In computer hardware, a regular signal must be used to trigger or start CPU work, the frequency of these signals is called clock rate, the higher the clock frequency, the faster the speed; The reciprocal is the clock period. The CPU can perform a very basic operation in each clock cycle.
* **The number of clock cycles per instruction (CPI) for a single instruction.** Program instructions are not necessarily basic actions in the CPU circuit, in fact, to complete an instruction, it usually takes several smaller execution cycles to complete; These execution cycles are driven by the clock signal provided by the hardware, so the length of the command can be expressed in clock cycles. However, different types of instructions may require different clock cycles, so we use its average value to represent the number of clock cycles of each instruction in the program, which is called the average number of clock cycles of a single instruction.
* The number of instructions (IC) in the program. From a programming perspective, instructions are the most basic units, because in the program structure, whether it is a **master(main) program**, **routine**, or **macro**, it is composed of a series of instructions. The number of times the program command is executed will inevitably directly affect the length of the program execution time.

**7 The performance of a P microprocessor with 100 MHz was evaluated by executing 10,000,000 evaluation program instructions, and it was found that it took 0.25 seconds. What are the CPI and MIPS data during this performance experiment?**

**Sol:**

(1) 

(2) 

8 The data in the following table was measured on a machine with a clock period of 2.5ns. Suppose the compiler designed by this machine can perform optimizations so that 50% of the ALU operations can be removed (but not other classes).

|  |  |  |
| --- | --- | --- |
| Command Class | Frequency of occurrence | Clock Cycles (CPI) |
| ALU ops | 40％ | 1 |
| Load | 25％ | 2 |
| Store | 10％ | 2 |
| Branch | 25％ | 2 |

1. What are the differences between MIPS when considering the compiler performing optimization and not performing optimization?
2. Are the results of MIPS comparison with these two different modes the same as the results of CPU time? Please give examples to substantiate your claims.

Sol:

(1)

CPI is not optimized：1×0.4 + 2×0.25 + 2×0.1 + 2×0.25 = 1.6

Unoptimized MIPS:

Optimize CPI：

Optimize MIPS:

**9 Suppose we have two processors using the same instruction set, where processor A has a clock period of 1ns and a CPI of 2.5, and processor B has a clock period of 2 ns and a CPI of 1.5 under the same program. How much faster?**

**Sol:**

(1) CPU Time = CPI × IC × Clock Cycle Time

Processor A: CPU Time = 2.5 × IC× 1ns = 2.5IC (ns)

Processor B: CPU Time = 1.5 × IC× 2ns = 3IC (ns)

Because the same instruction set is used, the IC part will be the same, so the processor A is faster.

(2) fasttimes

**10 (1) What is Anderson's theorem? According to this theorem, what do the x, y, z parameters represent in the following formula?**

****

**(2) If 20% of the system can use parallel processing to improve performance by 50%, what is the overall system growth rate after improving that part with parallel processing?**

**Sol:**

(1) Ander's theorem: When a local system is improved in a faster way to improve the overall system performance, the degree of improvement will be limited by the ratio of the "time it takes to actually run the local system" to the "overall system execution time" before the improvement.

x: The proportion of all work that is not the part that is not improved by the system.

y: The proportion of system improvements in all work.

z: After the system is improved, the performance of the improved part is a multiple of the growth rate compared to before the improvement.

(2)  (System growth rate 7.14%)

**11 There is a 500MHz processor, and the following is the data obtained after measurement**

|  |  |  |
| --- | --- | --- |
| Instruction set | CPI | frequency of use |
| A | 2 | 40% |
| B | 3 | 25% |
| C | 3 | 25% |
| D | 3 | 10% |

1. **What is the MIPS of this processor, please?**
2. **What is the CPI of this processor, please?**

**Later, the compiler performance of the processor was improved, and the following data was obtained**

|  |  |
| --- | --- |
| Instruction set | The improved percentage relative to the original frequency command |
| A | 90% |
| B | 80% |
| C | 85% |
| D | 90% |

1. **For the same program, what is the CPI if the improved compiler is used?**
2. **How much faster is it to use the improved compiler compared to the improved one?**

**Sol:**

(1) CPI = 

MIPS = 

(2) CPI = 2.6

(3) Improved CPI = 

(4) After MIPS improvement:

Therefore, it is twice as fast after improvement as before the improvement

**12 Please judge whether it is most cost-effective to buy a computer M1 or a computer M2, where the M1 clock is 1GHz and the M2 clock is 500MHz**

|  |  |  |
| --- | --- | --- |
| Instruction set | CPI for M1 | CPI for M2 |
| A | 4 | 2 |
| B | 6 | 4 |
| C | 8 | 3 |

1. **Compare the MIPS of the two processors.**
2. **If there are three compilers C1, C2, and C3, and the frequencies used are as follows**

|  |  |  |  |
| --- | --- | --- | --- |
| Instruction set | C1 | C2 | C3 |
| A | 30% | 30% | 50% |
| B | 50% | 20% | 30% |
| C | 20% | 50% | 20% |

* **If both computer M1 and computer M2 use compiler C1, who is faster? How much faster?**
* **Assuming that both computers can use the above three compilers, how to combine them to get the most benefits?**

Sol: (1) Assume that script sets A, B, and C appear at the same frequency:

M1 CPI=

M1 MIPS=

M2 CPI=

M2 MIPS=

So M1 MIPS = M2 MIPS

(2) M1 CPI = 

M1 MIPS = 

M2 CPI = 

M2 MIPS = 

Therefore, M1 is faster than M2, twice as fast

(3) Compiler C1:

M1 CPI = 

M2 CPI = 

Compiler C2:

M1 CPI = 

M2 CPI = 

Compiler C3:

M1 CPI = 

M2 CPI = 

Therefore, M1 and M2 will get high performance by using the C3 compiler.

If the MIPS is compared, then

M1 MIPS = 

M2 MIPS = 

Therefore, using M1 with C3 compiler can get the best benefits.

**13 What is the role of the Effectiveness Evaluation Process? What kind of problems may it face?**

**Sol:**

(1) Used to evaluate the processing performance of computer systems.

(2) Because the calculation mode of the evaluation program is too clear, many manufacturers of the tested system write compilers specifically for the characteristics of these programs in order to obtain better test data, such as deliberately "optimizing" the program into a very small machine code object, so that it is stored in cache memory almost all during execution, thus obtaining results that exceed normal speed. However, this approach is deceptive to testers, as the result data does not provide sufficient and accurate performance information, and the credibility of the evaluation program is questioned to a considerable extent.