Operating Systems

TEAM 6
Leonardo Nahra
Levindo Neto
Li-Wei Chen

Assignment 1

University of Stuttgart
06.11.2017

Assignment 1 - Answers

1.a)

Processor-memory: Instruction that involves transferring data between processor and memory.
Processor-I/O: Instruction that involves transferring data between processor and the I/O.
Data processing: Instruction that does arithmetic or logic computation on data.
Control: Instruction that can specify the execution sequence of a program.
Reference: Operating System Internal and Design Principle by William Stallings

1.b)

A processor typically uses two internal registers: MAR and MBR. A memory address register(MAR) assigned to the address in the memory for next read or write. A memory buffer register(MBR) stores the data that is read from memory or data to be written into the memory.

A processor also exchanges data with I/O devices using two registers: I/O AR and I/O BR. An I/O address register(I/O AR) specifies the I/O device for the next operation. An I/O buffer register(I/O BR) contains the exchange data between the CPU and the used I/O device.
Reference: Operating System Internal and Design Principle by William Stallings

1.c)

1. The program counter (PC) holds the address of next instruction for fetching it from memory.
2. The instruction is loaded to the MAR.
3. With the address in MAR, the data is loaded from main memory to MBR.
4. Copy the instruction from MBR to the instruction register(IR).
5. The PC increments by 1.
6. The instruction is then decoded and executed.

(The instruction is stored in the MBR. Considering the instruction will load a value from memory, it holds the value's address to be fetched from memory.
The value is stored in the MBR, and it may be used with the register as the ACC is, relying on the nature of the instruction.)
Reference: Lecture notes from Virginia Tech and University of Oxford

2.a)
Check whether the word is in the cache. If it is in the cache, it gets accessed directly. If the word is not in the cache, the processor will search the word in the next hierarchy' levels (next levels of cache, main memory, hard disk). After the word has been found, it will be copied to cache by the used replacement policy, and it will be delivered to the processor.
Reference: Lecture note from Uni. Hamburg

2.b)
0.6 * 1*ns* if the word is found in cache
+ (1 - 0.6)*(0.4)*(1 *ns* + 50 ns) if the word is found in memory
+ (1 - 0.6)*(1 - 0.4)*(1 *ns* + 50 *ns* + 1000 *ns*) if the word is in disk
= 261 *ns*

2.c)
Spatial locality is exploited by putting in cache a big block of memory that is near the data, which is being used. Temporal locality is exploited by keeping the recently used data in the cache. With these two locality principles, desired data and instruction are easily accessed in the cache with high probability, therefore the hit rate is improved.

3.a)
The process A has better CPU utilization because it frees up the CPU for doing the next instruction right after the sleep function. On the other hand, the process B has to enter in the Interrupt service routine, call sleep, then free the processor for doing other tasks. There is an idle time for CPU when process B enters in the ISR.

3.b)
If DMA access the main memory, it is given lower priority than the CPU, the reading or writing process may be paused by the processor, and the transferred data will be lost, as well as DMA does not fulfill the job of helping the CPU. In this case, it defies the purpose of having a DMA for handling a big block of data.
Reference: Operating System Internal and Design Principle by William Stallings

3.c)
The main goal of the dispatcher is to maximize the utilization of processor. It uses as few as possible instructions for deciding what is next instruction for being executed. Hence, there is a minimum idle time for the processor.
One way to achieve this is assigning a priority level for each process, and then, make a schedule based on the priority order (different scheduling methods may be used).

Tradeoff is a loss that can be gotten in order to achieve a gain. It can be understood as a deal between 2 parts, which one provides an improvement (e.g. performance) in exchange for a payment (use of memory).

Reference: Operating System Internal and Design Principle by William Stallings