**Chapter 1 Homework**

**Due:** Thursday, September 2, 2021, 11:59 PM

Honestly, this homework may seem tedious, but it should take you less time to complete than it took me to write it. It is not that hard. To make this as easy as possible I tried to keep the list of questions in the order the slides are in the slide deck.

This is an individual assignment.

**Submission:**

Submit this assignment through **Canvas.**

**Do not hand write your answers. Typing the answers makes grading much easier. You are required type your answers in RED. You are also required to show what slide number you found the answer on.**

To answer these questions, use the Chapter1.pdf, provided to you with this HW assignment..

1. Slide # \_\_\_2\_\_\_ How many bits are in 1 byte. 8 bits
2. Slide # \_\_\_5\_\_\_ In our discussions, we determined information is basically a bunch of bits and that a series of bits can represent different data. As an example, the binary number 00100011 can represent the decimal number 35 or it can represent the ascii value of 35, which is ‘#’. The thing that distinguishes different data is the \_context\_ in which we view the data.
3. Slide# \_\_\_6\_\_ What are the 4 phases of the compilation system?
   1. Pre-processor, compiler, assembler, linker
4. Slide# \_\_6\_\_\_ If you want to save all the temporary files created during the compile process what flag would you need to use when compiling?
   1. -save-temps
5. Slide# \_\_7-10\_\_\_ Assume you have a file hello.c and you compile this file using the flag to save the temporary files and you use the flag -o to name the executable **hello.** What temporary file will be created by the:
   1. Pre-processing phase \_hello.i\_
   2. Compiler phase \_\_\_hello.s\_\_\_
   3. Assembler phase \_\_hello.o\_\_
   4. Linker phase \_\_hello (the executable)\_\_
6. Slide# \_\_\_11\_\_ What does the compile flag -O0 mean?
   1. It stands for Optimization 0. This flag will set optimization to zero level. The compiler won’t automatically optimize your code during compilation process.
7. Slide# \_\_13\_\_\_ With respect to hardware organization of a system, what does CPU stand for? \_\_Central processing unit\_\_\_ We discussed three main areas of the CPU.

List them: \_\_\_ALU (arithmetic/logical unit)\_\_\_\_, \_\_\_\_PC (program counter)\_\_\_\_, \_\_\_\_USB (universal serial bus)\_\_

1. Slide# \_\_15\_\_\_ Describe what I/O devices are show in example we discussed? List the 4 that were given in our example. – I/O devices are input/output devices that are connected to the system from the outside world.
   1. Mouse
   2. Keyboard
   3. Display
   4. Disk drive
2. Slide# \_\_16\_\_\_ What is main memory?
   1. It is the temporary storage device that holds a program and the data it manipulates while the processor is executing the program.
3. Slide# \_\_17\_\_ \_\_CPU (Central processing unit)\_\_ is the engine that executes the instructions stored in main memory. It has a word size register called the \_\_PC (program counter)\_\_ that contains the address of some instruction in main memory.
4. Slide# 18 We discussed 4 operations that the CPU might carry out at the request of an instruction. Name and describe the 4 operations:
   1. Load – copies a byte or word from main memory into a register which overwrites the previous content of that register
   2. Store – copying a byte or a word from a register to main memory which overwrites the previous contents in that location in main memory.
   3. Operate – copy the elements of two registers to the arithmetic/logical unit (ALU), performs an arithmetic operation on the two words before storing the result in a register, which overwrites the previous data.
   4. Jump – extracting a word from the instruction and copying that word into the program counter, overwriting the previous value.
5. Slide# \_\_19\_\_With respect to running the hello world program, describe what is happening in the images below.
   1. As characters are typed into the command line they are being read as input into the USB controller from the keyboard. From there they go to the I/O bridge and are being read and saved into a register and then copied into main memory.

Timeline

Description automatically generated with low confidence

1. Slide# \_\_\_20\_\_ With respect to running the hello world program, describe what is happening in the images below.
   1. When the enter key is hit, the command is finished being defined. The executable file is loaded and stored on disk. Then using direct memory access, the code and data goes through the disc controller and I/O bridge and into main memory.

Timeline

Description automatically generated with medium confidence

1. Slide# \_\_21\_\_ With respect to running the hello world program, describe what is happening in the images below.
   1. Once the object file’s instructions are loaded in memory, the processor begins executing those instructions. The hello world string is then produced and copied into the registers and is then sent through the I/O bridge and to the graphics adapter where the hello world string is display on the output display.

Timeline

Description automatically generated

1. Slide# \_\_22\_\_ What was the motivation of developing Cache storage?
   1. The motivation for developing cache storage was to speed up processing times, as all the copying and moving information from one part of the computer to another takes a great deal of time.
2. Slide# \_\_24 and 25\_\_ The \_\_\_operating system (OS)\_\_\_\_\_ is a layer of software interposed between the application program and the hardware. This layer of software has 2 primary purposes. What are they:

a. Protect hardware from misuse by runaway applications

b. provide applications with simple and uniform mechanisms for manipulating complicated and different low-level hardware devices

1. Slide# \_\_26\_\_ With respect to an operating system, what is a process?
   1. A process is the operating system’s abstraction for running a program.
2. Slide# \_\_30\_\_ Starting at the zeroth address, list the 5 areas that make up virtual memory for Linux?
   1. Program code and data, heap, shared libraries, stack, and Kernel virtual memory
3. Slide# \_\_33\_\_ What are two demands that have been a constant force in driving improvements in computers and how can they be achieved?
   1. The two demands are wanting computers to do more and do more faster. These can be achieved with the processor doing more things at once using concurrency and parallelism.
4. Slide# \_\_35 and 36\_ What is a multi-core processor and describe how it is organized?
   1. A multi-core processor is a system of multiple processors that are all under the control of a single OS kernel. They are organized by having several CPUs (cores on a single chip. These cores then each have their own L1 and L2 caches. The L1 cache is divided into two parts with one part holding instructions and the other holding data.
5. Slide# \_\_38\_ Describe the listed 2 ways multiprocessing improves system performance.
   1. Multiprocessing improves system performance by reducing the need to simulate concurrency when performing multiple tasks and by running a single application faster if that program is expressed in terms of multiple threads that can execute in parallel.
6. Slide# \_\_39\_\_ Describe the two approaches to instruction level parallelism.
   1. The first approach is through hardware and dynamic parallelism, where the process decides at run which instructions to execute in parallel. The second approach is through software and static parallelism, where the compiler decides what instructions are executed in parallel.
7. Slide# \_\_41\_\_ When and who developed the “C” programming language?
   1. C was developed by Dennis Ritchie of Bell Labs from 1969 to 1973.