**Chapter 1**

**Compilation**

1. Give an example of a feature common in interpreted languages that is rare in compiled languages.

Interpreted languages allows codes to be run quickly as no compilation is required before running.

2) Name two advantages of static typing over dynamic typing.

Static typing allows bugs in the program to be identified even before the program runs.

Static typing allows the compiler/coder to enter and declare the correct type of data for each variable, which will reduce bugs/error.

With static typing, program can run faster as less steps will be used to check the data type during run time.

Easier for compiler to optimize

3) Give an example of a static semantic error.

Int a = “helloworld”;

A is declared as an integer but given a string (character array) since the data type doesn’t match, it will result in a semantic error.

4) What are two reasons you might want to turn off code optimization?

During the initial phase of writing and debugging your code, you might not want to add additional optimization which will **cause a longer compilation time**, and **likelihood of more errors** (which are due to the optimization). If you wish to get the basics (minimal) of your code up first, you might not want to turn optimization on, so that you don’t get unnecessary errors.

5) When you run gcc with -S, why might the results look different on different computers?

-S produces the assembly code, (.s) file. Since different computers run on different chips, architecture, syntax and structure of assembly code might vary. EG: ARM processor vs Intel processor have different assembly code.

6) If you spell a variable name wrong, or if you spell a function name wrong, the error messages you get might look very different. Why?

It is processed in different steps of the compilation.

A variable is processed during the “static checking” phase, where all the variables and their respective types are checked, so that bug can be found even before the code runs. It results in “static semantic error”

A function is processed during “Linking” where all the required functions/libraries are linked within to code.

7) What is a segmentation fault?

A segmentation fault happens when a pointer attempts to read from a deallocated memory address, or an address which is not supposed to be read. The address might be part of the operating system and hence cannot be read, resulting in segmentation fault. Example of such an error: when the pointer is not deleted or reallocated when you delete/deallocate an array/pointer etc.

**Chapter 2**

**Processes**

1. Give a real-world example of virtualization (ideally not one of the ones in the book).

Virtualization is the process of creating a desired illusion. An example of virtualization would be internet cloud storage (OneDrive, google drive etc.). One can purchase as much storage as you want, which can be accessed by anywhere with internet access. It creates an illusion that you always have the storage space, but it doesn’t matter if it is stored in Asia, America, or Europe (although the speed might be affected due to the distance). And you won’t have to care how the companies manages the storage.

1. What is the difference between a program and a process?

A program can consist of a number of process, each process have its own PID (process ID) which can help you identify, or kill the process etc.

Processes allow different components of the program, and also different programs, to be isolated from each other. Isolation prevents any undesired effect caused by other components. Programs are mainly stored in storage, while processes run in memory.

[Reference partly from: http://javaconceptoftheday.com/differences-between-program-vs-process-vs-threads/]

1. What is the primary purpose of the process abstraction? What illusion does the process abstraction create?

The purpose of process abstraction is so that users will not have to be affected by how the process works, but still able to utilize the functions of the program. It is to simplify something that can be complicated (at the back end). It will create an illusion that is simple and user-friendly.

1. What is the kernel?

Kernel is the core of a computer’s operating system which creates and manages computer resources such as CPU, threads, interrupts, memory, and peripheral devices (keyboard, monitor etc).

1. What is a daemon?

Daemon is a non-interactive process which runs in the background. It answers request for services.

Processes of daemon usually ends with ‘d’.

## Chapter 3

### Virtual memory

1. The Georgian alphabet has 33 letters. How many bit are needed to specify a letter?

6 bits. 2^5 =32 not enough to store 33 letters, hence 6 bits is required (can store up to 64)

1. In the UTF-16 character encoding, the binary representation of a character can take up to 32 bits.  
   Ignoring the details of the encoding scheme, how many different characters can be represented?

Maximum of 2^32 = 4294967296 different characters.

1. What is the difference between "memory" and "storage" as defined in Think OS?

Memory or Random Access Memory (RAM), are volatile, and memory is erased when it is power off. Transfer speed to/fro/within memory is faster than that of a storage. Processed are ran in memory.

Storage is where all the data/files(videos, music) are stored. Storage is non-volatile, means that the data remains even when power is off. Transfer speed of storage is slower than that of the ram. Programs are stored in storage, which then runs processes in memory when the computer powers on.

1. What is the difference between a GiB and a GB? What is the percentage difference in their sizes?

GiB stands for “gibibyte”, 2^30 bytes = 1073741824 Bytes ~~ 1.07 \*10^9 bytes

GB stands for gigabyte = 10^9 bytes

GiB/GB \*100 = 107%

% difference ~ 7%

1. How does the virtual memory system help isolate processes from each other?

Virtual memory system allows addresses within processes to be managed by the Memory management unit, which allocates each process a different chunk of memory to different processes (in some cases there might be some common memory to communicate within processes). This will isolate each process and prevents any undesired effects from a process interacting with other process’s address space.

1. Why do you think the stack and the heap are usually located at opposite ends of the address space?

Stacks grows down to a smaller address while heaps grow up to a larger address. Hence by locating them at opposite ends, they will have a larger “common” memory which they can expand to.

1. What Python data structure would you use to represent a sparse array?

Sorry, I’m not familiar with python. I understand sparse array as an array declared for maybe size of 100, but most of the time, maybe <10 are used. In C++, I might want to use a linked list so that memory is allocated only when the space is required. (I’m not very sure for the answer for this question). Or a hash table if the index of the data (in the array), is required.

8) What is a context switch?

Context switch is the process of interrupting a running process and save its state, so as to run another process. The state of the interrupted process can be resumed if required.

<Skipped the last part of CH3> will submit together with the next reflection