Chapter 5: Process Management in Theory

Introduction

Process is the term which is used in operating systems literature to point to a running program. In the previous chapters of this book we have encountered the concept of the process multiple times and you may recall from these encounters that every user-space software that we use in our computers are soulless sequence of bytes that are stored somewhere in the hard disk. When we decide to use a specific software, for example, the web browser, the first thing we do is to open it either through double click on its icon in graphical user interfaces or through writing its command in the shell. When we do that, the kernel is needed to be called and take the responsibility of "opening" this software, there are multiple steps that are needed to be performed to open the software, for example, reading its data from disk, but our current focus process-related parts, eventually, the kernel creates a new process for the software that we requested to open. After that, the kernel will have a table of current processes, each entry represents a process and contains the information which is needed by the kernel the manage the processes, this data structure which stores a process information is known as *process control block* (PCB). Of course, that most important part of the process is the code of the software that this process represents and its data, both data [1] and code should be loaded into memory, after that, its code can be executed by the processor. We need to note that a process is an instance of a software, in other words, one software can be opened more than one time, for example, opening multiple windows of the web browser on the same time, the software is one which is the web browser, but each opened window is a separated process.

[1] We mean static data here, which are contained in the file of the software. While the data that are generated by the running process are not loaded from the binary file, instead they are created while the code is running.