**Unit 1: Introduction to Secure Systems Architecture**

Welcome to Week 1. In this week we will introduce you to the concept of secure systems architectures, considered from the perspectives of operating systems and distributed systems, and the security challenges associated with each. You will be provided with a history of both operating and distributed systems, reinforced by the ways in which each have had to evolve due to changing user demands.

**On completion of this unit, you will be able to:**

* Understand the ways in which distributed systems have evolved to support evolutions in applications and user activities.
  + Explain how Operating Systems work and the purpose of processes, threads, and schedulers.
  + Describe how to set up and use OS Containers (e.g., Docker).

**Reflection:**

The **distributed computer system** shows how centralized systems shifted over period into decentralization and is thus all about development (van Steen and Tanenbaum, 2016).

An **operating system** is software that manages how other applications are run on a machine and handles communication among the user and the hardware. The OS is responsible for controlling and maintaining the computer's software and hardware components. It manages files, storage, and processes, processes input and output, and commands hardware like disc drives and scanners (www.uow.edu.au, n.d.). To the OS, whatever is now being carried out is known as a Process. For this reason, we use the term "Process" to refer to any running software. A browser is what you use to access the World Wide Web and conduct searches. Therefore, this is doable.

Using **synchronization**, program performance can be enhanced with the help of threads. A thread is a software implementation that is analogous to a traditional process. Threads are used to improve the speed of operating systems by lowering the amount of overhead they generate. There can be only one process associated with a given thread, and a thread cannot exist independently of a process.

The **scheduler's** job is to pick operations from a list of prepared activities, so that everything runs smoothly. The part of the Operating System known as the Dispatcher id is the part that is responsible for sending the prepared task to the processor to be performed (Williams, 2022).

**Containers** are a type of virtualization that can be used for operating systems. It's possible to utilize a single container to execute anything from a lightweight software process or microservice all the way up to a full-fledged app. Application container, such as Docker, enclose an application's files, requirements, and modules in a single package so that it can be run on an operating system. Application containers give the user the ability to design and run a distinct container for each of several applications that can run independently of one another or for several services that together make up a single application (Rubens, 2017).

**References:**

Rubens, P. (2017). What are containers and why do you need them? [online] www.cio.com. Available at: <https://www.cio.com/article/247005/what-are-containers-and-why-do-you-need-them.html>.

Van Steen, M. and Tanenbaum, A.S. (2016). A brief introduction to distributed systems. Computing, 98(10), pp.967–1009. doi:10.1007/s00607-016-0508-7.

Williams, L. (2022). Process Scheduling: Long, Medium, Short-Term Scheduler. [online] www.guru99.com. Available at: https://www.guru99.com/process-scheduling.html.

www.uow.edu.au. (n.d.). Understanding operating systems - University of Wollongong – UOW. [online] Available at: <https://www.uow.edu.au/student/learning-co-op/technology-and-software/operating-systems/#:~:text=The%20operating%20system%20(OS)%20manages>.