Multi-Programming Analogy

By: Anthony Sasso

March 21, 2022

Revision History

Revision	Date	Author(s)	Description
1.5	2022/03/21	Anthony Sasso	converted to .tex from .rtf, changed formatting but no core text altered yet
1	2021/02/25	Anthony Sasso	original rtf this is from lists last modification at this time (not technically possible but didn't use version history originally. . .)
0.5	2021/09/30	Anthony Sasso	created

Contents

Introduction / Concept

Just an analogy I had said in class for understanding multiprogramming terms that I thought I should write down for later. . .

Essentially "One way to understand the differences between processes, threads, semaphore, and mutex is media like Azure / GitHub, Google Docs, email, and a punch out card.".

Multiprocessing

Multiprocessing operates fundamentally separately with the fork() command creating a segmented process with identical data (which can then merge / push using the pipe() command). This is relatable to how Azure and GitHub creates a clone / local copy that can be manipulated until it is merged back, overwriting the data in the parent or main directory.

Multi-Threading

Threads somewhat similarly are shared data with multiple "actors" manipulating and altering the information. This can be seen analogously in 'Google Docs' with multiple people all writing to the same document, with each of them being unique, separately operating functions with a shared resource pool between them.

Semaphores

A semaphore differently can only send a "flag" to other linked processes potentially updating, altering, or alerting them. Think of this as an employee waiting for their boss to give them the go-ahead email, not able to continue until this is reached; or that same employee receiving notification they must stop production of a new product immediately.

Mutexes

Mutexes are the "hard-coded" alternative of semaphore. Think of it like a punch out card where the halted program cannot and will not continue until another program (the one who initiated the mutex) "punches out" and lets it continue.