VE482 - Homework 3

Introduction to Operating Systems

Kexuan Huang 518370910126

October 19, 2021

Ex.1 General Questions

Why would a thread voluntarily release the CPU?

Threads work together for the task in the same process to achieve more optimal outcomes as programmers' need. As a result, if one thread's voluntary release of the CPU is needed for the good, then we programmers code it as so.

What is the biggest advantage/disadvantage of user space threads?

The biggest advantage: efficiency. No traps to the kernel are needed to switch threads; User space threads are easier and faster to create than kernel-level threads, as well as be more easily managed; User space threads can be run on any operating system.
The biggest disadvantage: The entire process is blocked if one user space thread blocks.

If a multithreaded process forks, a problem occurs if the child gets copies of all the parent's threads. Suppose that one of the original threads was waiting for keyboard input. Now two threads are waiting for keyboard input, one in each process. Does this problem ever occur in single-threaded processes?

No, this problem doesn't occur in single-threaded processes. Because if a single-threaded process is blocked and waiting the keyboard input, it cannot fork.

Many UNIX system calls have no Win32 API equivalents. For each such call, what are the consequences when porting a program from a UNIX system to a Windows system?

For calls having no Win32 API equivalents, chances are that they don't run properly when being ported from a UNIX system to a Windows system.

Ex.2 C programming

See `./src` folder and `README.md` for more detail.

Ex.3 Research on POSIX

The Portable Operating System Interface (POSIX) is a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems.

POSIX was supposed to make it easier to write cross-platform software, which defines both the system-level and user-level application programming interfaces (API), along with command line shells and utility interfaces, for software compatibility (portability) with variants of Unix and other operating systems.

The POSIX specifications for Unix-like operating systems originally consisted of a single document for the core programming interface, but eventually grew to 19 separate documents (POSIX.1, POSIX.2, etc). The standardized user command line and scripting interface were based on the UNIX System V shell. Many user-level programs, services, and utilities (including awk, echo, ed) were also standardized, along with required program-level services (including basic I/O: file, terminal, and network). POSIX also defines a standard threading library API which is supported by most modern operating systems.