VE482 Homework 3

Lan Wang 519370910084

Ex. 1 — General questions

1. **Why would a thread voluntarily release the CPU?**

   If the application needs threads to release the CPU, they will voluntarily release the CPU, because this is wrote by programmers.

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

   advantage: efficiency; no traps to the kernel are needed to switch threads

   disadvantage: if one thread blocks, the entire process blocks

3. **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. For single-threaded process, it will be blocked and cannot fork.

4. **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?**

   They will be changed into combinations of Win32 APIs and other codes.

Ex. 2 — C programming

See attached files.

Ex. 3 — Research on POSIX

The Portable Operating System Interface (POSIX) is an IEEE standard It helps compatibility and portability between operating systems. Theoretically, POSIX compliant source code should be seamlessly portable. In the real world, application transition often runs into system specific issues, but POSIX compliance makes it simpler to port applications which can save time.

It includes:

- POSIX Base Definitions
- System Interfaces
- Commands and Utilities
  - POSIX.1 & extensions for POSIX.1
  - Real-time Services
  - Threads Interface
  - Real-time Extensions
  - Security Interface
  - Network File Access and Network Process-to-Process Communications
  - User Portability Extensions
  - Corrections and Extensions
  - Protection and Control Utilities and Batch System Utilities

Referred to: https://en.wikipedia.org/wiki/POSIX