

Ryan Roelke

Capstone Abstract

Weenix

Intended to construct a better understanding of operating systems through implementation, Weenix is a simple operating system written in C. The project consists of five principal components: processes, threads and scheduling; disk and terminal device drivers; a virtual file system; an on-disk file system implementation (S5FS); and virtual memory. Each of these components built upon the previous, culminating in a kernel capable of executing programs stored on disk in user mode. Consequently, uniting each of these components at the end of the project often revealed problems with code written months before. To solve such issues, understanding each part of the project and its interactions with the other parts was critical. Resolving the most difficult challenges usually required examining the smallest unit of kernel data structures, including lists of processes and threads, individual blocks of the disk, and virtual memory mappings of a process.