

Operating System Support for Programming Languages

Topics

- System Call Overview
- I/O
- Memory Allocation/Deallocation
- Dynamic Linking
- Error/Exception Handling
- Thread Support
- Communication between devices

System Call Overview

By Robert Valenti

System Call Overview

- Request to operating system
- Calls
 - Originate in user space
 - open, read, write, close, wait, fork, exit
 - Linux 300+

Steps of a System Call

- Computer saves its state
- OS controls CPU, privileges are updated.
- OS checks call parameters
- OS performs the requested function
- OS saves state
- OS gives CPU control to caller

Categories of System Calls

- Process Control
 - Fork on Unix
 - NtCreatePRocess on Windows
- File management
 - Creating/Deleting/Opening/Closing a file
- Device Management
 - Request device
- Information Maintenance
 - Get certain data, time
- Communication
 - Sending and receiving messages

Multiple Systems

- Avoid calling system calls in portable code
 - Linux/Windows Calls
- Programming languages usually provide some wrapper / api for the system calls
 - C runtime library
 - C = fopen, fread, fwrite, fclose
 - fopen = open

System Call Overview

- API calls via system call interface
- APIs instead of direct system calls

Sources

- <http://www2.cs.uregina.ca/~hamilton/courses/330/notes/unix/filesyscalls.html>