第 5 讲：The Interface of OS

第三节：POSIX in modern OS

陈渝

清华大学计算机系

yuchen@tsinghua.edu.cn

2020 年 3 月 15 日

Introduction

POSIX Abstractions in Modern Operating Systems: The Old, the New, and the Missing

Vaggelis Atlidakis, Jeremy Andrus, Roxana Geambasu, Dimitris Mitropoulos, and Jason Nieh

Columbia University

Goals

- Study the evolution of abstractions in modern OSes
- Understand how modern workloads use traditional abstractions
- Identify the needs of modern applications

Questions

- Which POSIX abstractions are unpopular for modern apps?
- Which POSIX abstractions are popular for modern apps?
- Is POSIX missing any functionality?

POSIX Abstractions in Modern Operating Systems: The Old, the New, and the Missing, Vaggelis Atlidakis, etc., EuroSys, 2016
https://columbia.github.io/libtrack/

Introduction – Workloads & Methodology

Three Modern OSes

- Android 4.3, Ubuntu 12.04 , and OSX 10.10

Client-side Apps

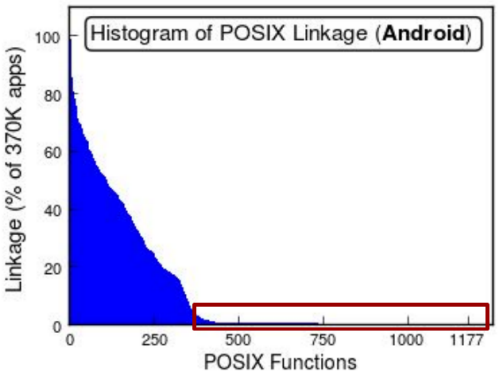
- Facebook, Twitter, Skype, Chrome, Safari

Common User Workloads

- e.g., post update, tweet, video call, browse

Static & Dynamic Measurements

Q1: Which POSIX abstractions are unpopular for modern apps?



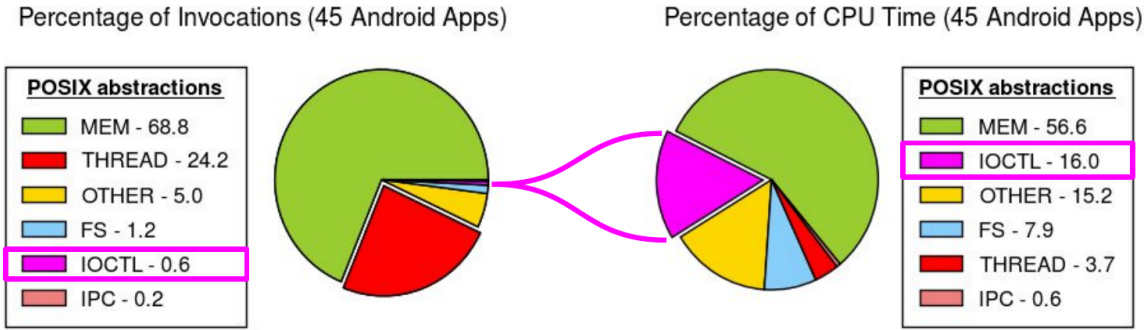
Long tail of unused interfaces

IPC (only 32% implemented in Android)

- No shared_mem, mq
- Partially pipes, semaphores
- Very few apps link to mkfifo

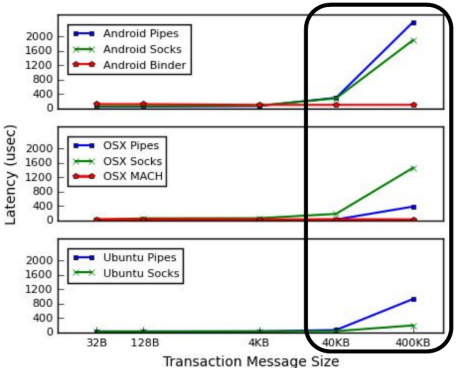
Large numbers of unused or unimplemented abstractions, Departure from traditional IPC and async I/O

Q2: Which POSIX abstractions are popular for modern apps?



IOCTL: Extension API used to shortcut POSIX; Directly interact with the kernel; Build functionality not expressed from POSIX APIs

Study Questions

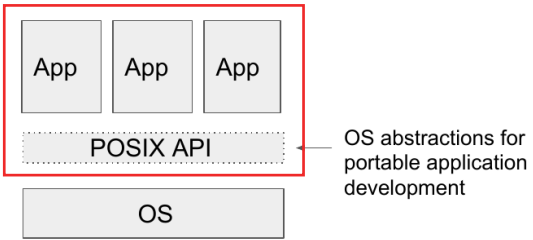


Q3: Is POSIX missing any functionality?

- POSIX omits graphics abstractions
- OpenGL cross-platform API used by applications
- No standard interface to GPUs but ioctl
- Binder IPC is a central abstraction in Android
- Similar patterns in other OSes (MACH IPC, D-Bus)
- GUI apps require low-latency UI threads

Evolution of systems and applications

In the past



Now