

Contents

| | | |
|----------|--------------------------------------|----------|
| 1 | Introduction | 7 |
| 1.1 | System programming Concept | 7 |
| 1.2 | System Call | 8 |

Todo list

| | |
|---------------------------|---|
| Linux layer | 7 |
| kernel mode | 7 |
| kernel function | 7 |

Abstract

The goal of this reader is to summarize my learning in system programming in `C`.

Chapter 1

Introduction

1.1 System programming Concept

- What is a kernel?

The kernel is the core of an OS, which handles resources.

It resides inside the memory, and it is the 1st thing that comes at the start of a bootloader

- Function of a kernel

- file management
- memory management: making the memory available to different processes via the virtual memory management
- Interprocess communication: how different processes communicate with each other
- Threads (so multitasking system)

- Linux as layer view

- Linux hide hardware from direct access by user application, because if so it can damage the hardware
- Linux layer: to insert a picture later

Linux layer

- When accessing hardware, the OS turns into kernel mode

- kernel mode: to see if this is called also privilege mode

kernel mode

kernel functions and concept

kernel funct

- *to define and write later different new concept used, like what is a process, a thread, concepts about Linux OS, ...*
- *Add references about system programming later*

1.2 System Call

- There are 2 ways to perform system call:
 - Through a library function.
Example is `printf()`
 - Or using directly the kernel like the function `write()`
- Every system call has a unique number associated with it
- The functional block diagram is shown in [Figure 1.1](#).

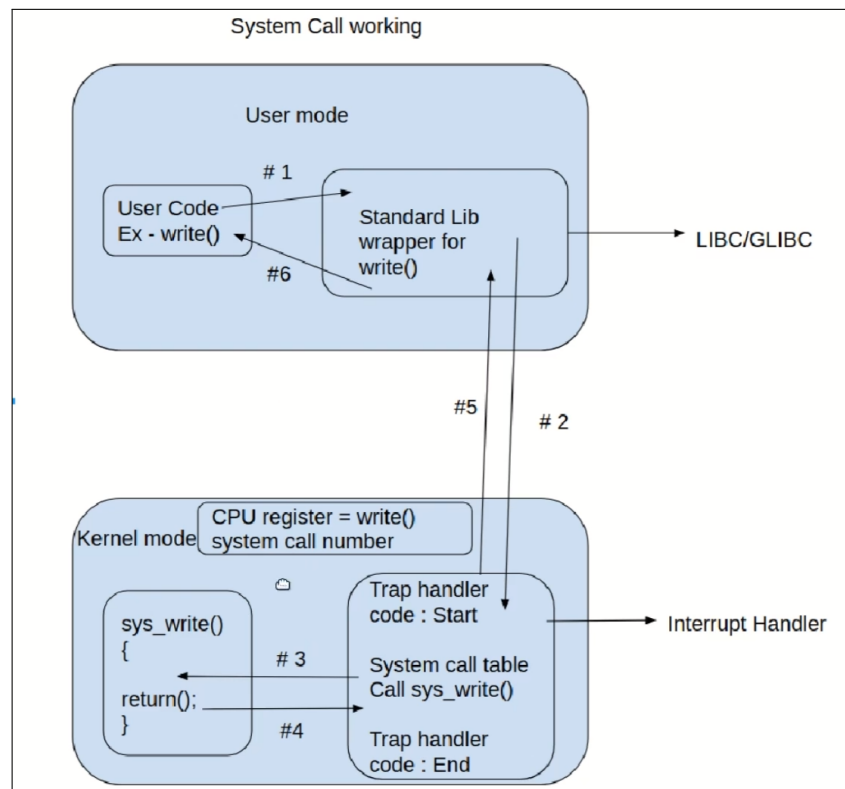


Figure 1.1: Main Components

•