System Call

Operating System

Juhi Maheshwari (121020)[1] Kashyap Patel (121021)[1]

[1]Institute of Engineering and Technology Ahmedabad University

Final Presentation, 2015

Outline

Introduction

System calls

- Introduction
- Categories
- Need of System call
- Diffrence : System call V/S Library function

Adding System call to Linux Kernel

- Introduction
- kernel compilation
- Testing

Conclusion

Abstract

Project Abstract

- The basic problem statement is to write a system call.
- The approaching step is to compile and install the latest version of kernel. Followed by, writing a system call and adding it to kernel.
- Then kernel needs to be recompiled to test the system call function. A simple C program has to be written to invoke our system call.

Project Division

Project Mainly divided into 3 part

- Downloading and initialization subversion repository with a copy of recent stable version
- Creation of new branch in that repository within that branch and then write a system call and register it with kernel
- This part is testing part. In this part we compile and install our modified kernel and then write a simple C code to invoke system call.

Outline

Introduction

System calls

- Introduction
- Categories
- Need of System call
- Diffrence : System call V/S Library function

Adding System call to Linux Kernel

- Introduction
- kernel compilation
- Testing

Conclusion

Blocks

What is System call?

System call is the fundamental interface between application and the Linux kernel. System call provides an essential interface between a process and the operating system.
A system call is an entry point into the Linux kernel. Usually, system calls are not invoked directly: instead, most system calls have corresponding C library wrapper functions which perform the steps required in order to invoke the system call.
so making a system call looks the same as invoking a normal library function.
System calls can be invoked from userspace process as well as other system call can also call the required system call.

Outline

Introduction

System calls

- Introduction
- Categories
- Need of System call
- Diffrence : System call V/S Library function

Adding System call to Linux Kernel

- Introduction
- kernel compilation
- Testing

Conclusion

Categories of System call (1/2)

Process Control

Process control is basically changing the state of process. Running process, if invokes an I/O request must be blocked or put in a waiting state.

File Manipulation

Some common system calls are create, delete, read, write and close.

Device Manipulation

Process usually requires several resources to execute. If these resources are available, they will be granted and control returns to the user process. Resources are released by process as soon as it terminates or has finished executing.

Categories of System call (2/2)

Information Maintenance

Some system calls exist only to transfer information between user program and kernel/OS.

Communication

It refers to inter-process communication. It can be done by message passing technique. Message-passing uses a common mailbox to pass messages between processes.

Outline

Introduction

System calls

- Introduction
- Categories
- Need of System call
- Diffrence : System call V/S Library function

Adding System call to Linux Kernel

- Introduction
- kernel compilation
- Testing

Conclusion

Need of System call

- System calls acts as entry point to OS kernel.
- There are certain tasks that can only be done if a process is running in kernel mode.
- System calls are also used for switching the execution mode. Example : systenter sysexit

Outline

- Introduction
- System calls
  - Introduction
  - Categories
  - Need of System call
  - Diffrence : System call V/S Library function
- Adding System call to Linux Kernel
  - Introduction
  - kernel compilation
  - Testing
- Conclusion

System call V/S Library function

- A library function is linked to the user program and executes in user space while a system call is not linked to a user program and executes in kernel space.
- Library functions execution time is counted in user level time while a system call execution time is counted as a part of system time.
- Library functions can be debugged easily using a debugger while System calls cannot be debugged as they are executed by the kernel.

Outline

Introduction

System calls

- Introduction
- Categories
- Need of System call
- Diffrence : System call V/S Library function

Adding System call to Linux Kernel

- Introduction
- kernel compilation
- Testing

Conclusion

Introduction(1/2)

- Kernel version comes with Many System calls but you can create your own as well

Steps to follow

- Extract the kernel source code at /usr/src location
- Go to Kernel directory

Define New System call

Now you have to define your own System call.

- make new directory
- Add definition of System call in that directory

Introduction (2/2)

- Create Makefile
- Add directory into Kernels Makefile
- Add the new system call into the system call table
- Add the new system call in the system call header file.

Outline

Introduction

System calls
- Introduction
- Categories
- Need of System call
- Diffrence : System call V/S Library function

Adding System call to Linux Kernel
- Introduction
- kernel compilation
- Testing

Conclusion

Kernel Compilation

Now next step is to Compile kernel.
steps :

- Go to Kernel Directory
- make configuration
- Do make
- Make modules
- install Modules
- install Kernel

Outline

Introduction

System calls

- Introduction
- Categories
- Need of System call
- Diffrence : System call V/S Library function

Adding System call to Linux Kernel

- Introduction
- kernel compilation
- Testing

Conclusion

Testing of System call

Testing

A c program, calling a system call from Syscall64.tbl file is created and executed.The output needs to be checked in kernel logfile.

Work done till date

Given Project

As mentioned before, the project was divided into 3 stages. All of those stages were completed long before.

Additional Work

Then the task was to implement a buffer of size 5 as a system call. So whenever any other process needs a buffer, the only task would be to call that system call and buffer would be assigned. The system call returns the head position of the buffer. Hence a permanent buffer is made in system. So whenever needed you dont to make it explicitly.

Conclusion

Future Work

Implementing producer consumer problem using the buffer implemented in system call. As producer produces the item, it gets stored into buffer and consumer can consume from that buffer.

Learning

- Interaction With Kernel
- writing a system call and calling it from application layer through c code.
- understanding of kernel, kernel libraries, kernel functions and its structure.
- We learnt how to add system call to kernel and how to compile it afterwards.

Reference

- http://www.tldp.org/LDP/lkmpg/2.4/html/x939.html
- http://en.wikipedia.org/wiki/Systemcall
- http://www.thegeekstuff.com/2012/07/system-calls-library-functions/
- http://man7.org/linux/man-pages/man2/intro.2.html
- http://faculty.salina.k-state.edu/tim/ossg/Introduction/sys$_{c}alls.html$