System call Implemention for Fedora -Operating-System

What is a System Call?

A system call is a method used by user-space programs to interact with the operating system kernel. It provides a controlled way for programs to request services like file operations, process management, memory handling, and more.

Why Implement a System Call?

Implementing a system call helps you understand how the Linux kernel handles user-space requests, Learn kernel development and system-level programming , Explore security and performance inside the OS.

How to Implement a System Call in Fedora (Linux Kernel)

Step 1: Prepare the Environment

Use a virtual machine (e.g., VMware or VirtualBox) to avoid damaging your main OS.

Install necessary packages:

sudo dnf groupinstall "Development Tools" sudo dnf install

ncurses-devel elfutils-libelf-devel bc openssl-devel wget

Step 2: Download and Configure Kernel Source

Download the Fedora kernel source: sudo dnf install fedora-repos-rawhide

sudo dnf builddep kernel rpm -q --qf "%{VERSION}-

%{RELEASE}.%{ARCH}\n" -f /boot/vmlinuz-$(uname -r) sudo dnf download

--source kernel

Extract and enter the kernel source folder:

rpm2cpio

kernel*.src.rpm | cpio -

idmv tar -xvf linux-

.tar.xz cd linux-/

Step 3: Add Your Custom System Call

Create a system call function in kernel/sys.c: asmlinkage long sys_helloworld(void) {

printk(KERN_INFO "Hello, Fedora World!\n");

return 0;}

Add the prototype to include/linux/syscalls.h:

asmlinkage long sys_helloworld(void);

Assign syscall number: Edit arch/x86/entry/syscalls/syscall_64.tbl:

548 common helloworld

__x64_sys_helloworld Step 4:

Recompile the Kernel make

menuconfig # Optional configuration

make -j$(nproc) # Compile sudo

make modules_install sudo make

install Update GRUB and reboot: sudo

grub2-mkconfig -o

/boot/grub2/grub.cfg sudo reboot

Step 5: Test Your System Call

Create a test program:

#include <stdio.h>

#include <unistd.h>

#include

<sys/syscall.h>

#define

__NR_helloworld

548 int main()

{ long status =

syscall(__NR_helloworld);

printf("System call returned: %ld\n",

status); return 0;}

Compile and run: gcc test.c -o test ./test

Check kernel logs: dmesg | tail