Assignment No: 9

Title: System Call

AIM:

Implement a new system call, swipe(), in the Linux kernel that transfers the remaining time slice of each process in a specified set to a target process. Demonstrate various uses of the system call.

OBJECTIVE:

To create and integrate the swipe() system call into the Linux kernel, and illustrate its effects both beneficially and detrimentally.

THEORY:

Steps to Add a New System Call:

1. Download the Kernel Source:

bash

wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.17.4.tar.xz

2. Extract the Kernel Source Code:

bash

sudo tar -xvf linux-4.17.4.tar.xz -C /usr/src/

cd /usr/src/linux-4.17.4/

3. Create the System Call Implementation:

Navigate to the kernel source directory and create a new directory for your system call code:

mkdir -p /usr/src/linux-4.17.4/my_syscalls

Then, implement the swipe() function in a new C file, e.g., swipe.c, inside this directory.

4. Modify Kernel Files to Register the System Call:

Edit the necessary kernel files to register the new system call:

- Open arch/x86/entry/syscalls/syscall_64.tbl and add a new entry:

  [syscall_number] 64 swipe   my_syscalls/swipe

- In include/linux/syscalls.h, add a function prototype:

  asmlinkage long sys_swipe(pid_t target_pid, pid_t *process_set, int num_procs);

5. Configure the Kernel:

Use the following commands to configure the kernel:

```
make menuconfig
```

Ensure the new module is included if you have added a CONFIG option for it.

6. Compile and Install the Modified Kernel:

Run the following commands to build the kernel and install it:

```
make -j$(nproc)
sudo make modules_install
sudo make install
```

Reboot the system with the new kernel version.

7. Test the New swipe() System Call:

After booting with the new kernel, you can write a user-space program to test the swipe() system call. Use the syscall() function in your C program to call swipe() and observe the results based on the target process and process set parameters.