

ASSIGNMENT 10

Problem Statement: Implement a new system call add this new system call in LINUX Kernel (any kernel source, any architecture and any Linux Kernel distribution) and demonstrate use of some.

Theory :

Adding a Simple System Call

Simple Example Say we wanted to add our own version of system
call getpid (). Lets call our version mygetpid ().
The implementation of my getpid () is

æmlinkage long sys = getPid (void)

return current > tgid;

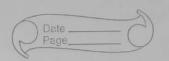
assemlinkage must appear before every system call.

It tells the compiler to only look on the stack

for function argument

Steps needed to follow to add this system Call-

- 1) Implement punction (all and put it in appropriate file.
- 2) Add an entry to end of system call table.
- 3) Define system call number in include / asm/unistdoh
- 4) Recompile your kernel and boot to it.



Accessing the system call from User - Space -

Compile Command -

gcc - I / The / Location / Of / your / linux / include test syscall · c

Adding swipe () system call

Swipe () that steads the collective time slice for a

Specified set of processes and adds it to target

process timeslice. The swipe () system call takes a

target process id, an integer that provides type of

set of processes and an integer that provides the

process set.

The system call returns the total amount of the timeslice swiped on a negative number if an error occured. Therefore, the prototype of function is:

int swipe (Pid -t target, int which, int who)

Using Swipe ():-

1) Create a process that monopolizes the (PU 2) Create a wrapper program that takes a simple command starting with the collective timeslices Of all processes in the set

I have successfully understood the concept of new system call.