This is a new system call implementation to print process tree with there child’s.

To use this call make following changes.

1. Download new linux kernel from kernel.org

-In your terminal type the following command:

wget https://www.kernel.org/pub/linux/kernel/v5.x/linux-5.7.4.tar.xz

2. Extract the kernel source code

sudo tar -xvf linux-5.7.4.tar.xz -C /usr/src/

3. cd /usr/src/linux-5.7.4/

4. mkdir print\_proc\_tree

5. cd print\_proc\_tree

6. Copy print\_proc\_tree.c and Makefile file's here.

7. cd ../

8. Open Makefile and change the line core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/ TO core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/ print\_proc\_tree/

9. Add the new system call to the system call table.

-cd arch/x86/entry/syscalls/

-gedit syscall\_64.tbl

-Go to the last of the document and add a new line like so:

-548 64 print\_proc\_tree sys\_print\_proc\_tree

-Save changes and exit

10. Add new system call to the system call header file:

-cd include/linux/

-gedit syscalls.h

-Add the following line to the end of the document before the #endif statement:

-asmlinkage long sys\_print\_proc\_tree(int pid);

-Save changes and exit.

-Some architectures (e.g. x86) have their own architecture-specific syscall tables, but several other architectures share a generic syscall table. Add your new system call to the generic list by adding an entry to the list in **include/uapi/asm-generic/unistd.h**

-#define \_\_NR\_ print\_proc\_tree 548

\_\_SYSCALL(\_\_NR\_print\_proc\_tree, sys\_print\_proc\_tree)

-Save changes and exit

- The file **kernel/sys\_ni.c** provides a fallback stub implementation of each system call, returning -ENOSYS. Add your new system call here too

-COND\_SYSCALL(sys\_print\_proc\_tree);

-Save changes and exit.

11. Compile/ Install / update Kernel

-to configure your kernel use the following command in your linux-5.7.4/ directory:

-sudo make -j 4 && sudo make modules\_install -j 4 && sudo make install -j 4

-Once this is done, restart the system.

12. Testing the system call

-Download main.c file and build it.

-Run ./a.out <pid>

-Run dmesg ### Check the kernel log to which we print the process info