

Assignment 2

| **Student:** Alicia Rodriguez - 5162522

| **Due Date:** 06/12/2017 by 11:55pm

What I Learned

I learned how to do the following things:

- Create a static system call and implement it in the linux kernel.
- Create a Makefile.
- Use the `task_struct` struct to get information about the current process running.
- Create a shell script and run it.

Additionally, this assignment has enhanced my skills with the C programming language and kernel programming.

Add a Static System Call to the Linux Kernel

1. Added `.long sys_pinfo` at the end of the `/usr/src/linux/arch/i386/kernel/syscall_table.S` file.
2. Since line 223 is not in use in the file `/usr/src/linux/include/asm-i386/unistd.h`, `#define __NR_pinfo 223`, was add.
3. Added the same define line in #2 to `/usr/src/linux/include/asm-x86_64/unistd.h`, except it is on 301.
4. In `/usr/src/linux/include/linux/syscalls.h` added 3 things.
 - a. `struct pinfo`; to reference the pinfo struct created.
 - b. `#include <linux/pinfo.h>` to include the pinfo struct header file
 - c. `asmlinkage long sys_pinfo(struct pinfo *info)`; to add the system call prototype
5. Updated `/usr/src/linux/Makefile` by adding the `/pinfo` directory created. `core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/ pinfo/`

KERNEL COMPONENTS MODIFIED:

- `/usr/src/linux/arch/i386/kernel/syscall_table.S`

- /usr/src/linux/include/asm-i386/unistd.h
- /usr/src/linux/include/asm-x86_64/unistd.h
- /usr/src/linux/include/linux/syscalls.h
- /usr/src/linux/Makefile

KERNEL COMPONENTS CREATED:

- /usr/src/linux/include/linux/pinfo.h
- /usr/include/pinfo.h
- /usr/src/linux/pinfo
- /usr/src/linux/pinfo/pinfo.c
- /usr/src/linux/pinfo/Makefile

REFERENCES:

- Implementing a System Call on Linux 2.6 for i386 <http://tldp.org/HOWTO/html_single/Implement-Sys-Call-Linux-2.6-i386/#AEN19>
- PRIO_TO_NICE Macro Implementation <<http://elixir.free-electrons.com/linux/latest/source/include/linux/sched/prio.h#L6>>

Testing the System Call

The following fields in the pinfo structure change (almost every time the test program is executed):

	Field	Why it changes
1	pid	Since there are different processes running, they are assigned different process IDs.
2	start_time	The start time of the program will be different every time it has been executed since the system has been active.
3	user_time	This changes occasionally when the process is being run in user-space.
4	sys_time	This changes occasionally when the process is being run in kernel-space.
5	nice	The nice value changes frequently because this essentially sets a CPU priority that the kernel scheduler will use to determine which processes get more or less CPU time.
6	parent_pid	This changes for the same reason the pid changes.
7	youngest_child_pid	This changes for the same reason the pid changes. This is essentially the last node in the children linked list.

	Field	Why it changes
8	younger_sibling_pid	This changes for the same reason the pid changes. This is essentially the first node in the sibling linked list.
9	older_sibling_pid	This changes for the same reason the pid changes. This is essentially the last node in the sibling linked list.
10	cutime	This changes when the total user time of children has changed.
11	cstime	This changes when the total user time of children has changed.
12	nr_children	This changes depending on the amount of child processes that are created in the test program. These will be incrementally changing up to the maximim number of child processes because they are being created in a for loop 1 at a time.
13	nr_threads	This changes depending on the amount of threads that are created in the test program.
14	state	This field will only change if the program is either, 0 runnable, -1 unrunnable, >0 stopped. It displays 0 all the time because the program is runnable.

The following fields in the `pinfo` do not structure change:

	Field	Why it doesn't change
1	comm	This is the name of the program, therefore it will not change. unless the name of the program has changed.
2	uid	This will change depending on what user account the program is being executed on. for root account it is always 0.

TEST FILES CREATED:

- `test_pinfo.c`
- `siblings.sh`

Test program needs to be executed the following way: `gcc -pthread test_pinfo.c -o test`

SAMPLE OUTPUT

```
[[root@localhost Assignment2]# ./siblings.sh

PID      STATE   NICE    P_PID   NR_C    NR_T    YCPID   YSPID   OSPID   STIME   UTIME   SYTIM   CUTIM   CSTIM   UID     COMM
3663      0        1       3660    1        3       3664    3665    3662    1093    0        1        0        0        0       test
3663      0        3       3660    2        3       3670    3666    3662    1093    0        1        0        0        0       test
3663      0        5       3660    3        3       3674    3666    3662    1093    0        1        0        0        0       test
[[root@localhost Assignment2]# ./siblings.sh

PID      STATE   NICE    P_PID   NR_C    NR_T    YCPID   YSPID   OSPID   STIME   UTIME   SYTIM   CUTIM   CSTIM   UID     COMM
3687      0        0       3684    1        3       3689    3688    3686    1320    1        1        0        0        0       test
3687      0        3       3684    2        3       3694    3688    3686    1320    1        1        0        0        0       test
3687      0        5       3684    3        3       3698    3688    3686    1320    1        1        0        0        0       test
[2] Done sleep 15
[4]- Done sleep 30
[5]+ Done sleep 15
[[root@localhost Assignment2]# ./siblings.sh

PID      STATE   NICE    P_PID   NR_C    NR_T    YCPID   YSPID   OSPID   STIME   UTIME   SYTIM   CUTIM   CSTIM   UID     COMM
3705      0        1       3702    1        0       3706    3707    3704    1333    1        2        0        0        0       test
3705      0        1       3702    2        3       3711    3707    3704    1333    1        2        0        0        0       test
3705      0        4       3702    3        3       3715    3707    3704    1333    1        2        0        0        0       test
[[root@localhost Assignment2]# ./siblings.sh

PID      STATE   NICE    P_PID   NR_C    NR_T    YCPID   YSPID   OSPID   STIME   UTIME   SYTIM   CUTIM   CSTIM   UID     COMM
3723      0       -2       3720    1        3       3725    3724    3722    1341    1        1        0        0        0       test
3723      0        5       3720    2        0       3730    3724    3722    1341    1        1        0        0        0       test
3723      0        5       3720    3        3       3734    3724    3722    1341    1        1        0        0        0       test
[[root@localhost Assignment2]# ./siblings.sh

PID      STATE   NICE    P_PID   NR_C    NR_T    YCPID   YSPID   OSPID   STIME   UTIME   SYTIM   CUTIM   CSTIM   UID     COMM
3741      0        2       3738    1        1       3742    3746    3740    1343    0        1        0        0        0       test
3741      0        4       3738    2        3       3747    3745    3740    1343    0        1        0        0        0       test
3741      0        5       3738    3        3       3751    3745    3740    1343    0        1        0        0        0       test
[[root@localhost Assignment2]# ./siblings.sh

PID      STATE   NICE    P_PID   NR_C    NR_T    YCPID   YSPID   OSPID   STIME   UTIME   SYTIM   CUTIM   CSTIM   UID     COMM
3759      0        2       3756    1        3       3760    3761    3758    1344    1        1        0        0        0       test
3759      0        4       3756    2        3       3765    3761    3758    1344    1        1        0        0        0       test
3759      0        5       3756    3        3       3769    3761    3758    1344    1        1        0        0        0       test
```