CSCE 326 Assignment 3

Due: March 29, 2019 by class time on BeachBoard

Goals

1- To practice basic kernel programming
2- To enhance the understanding of virtual memory

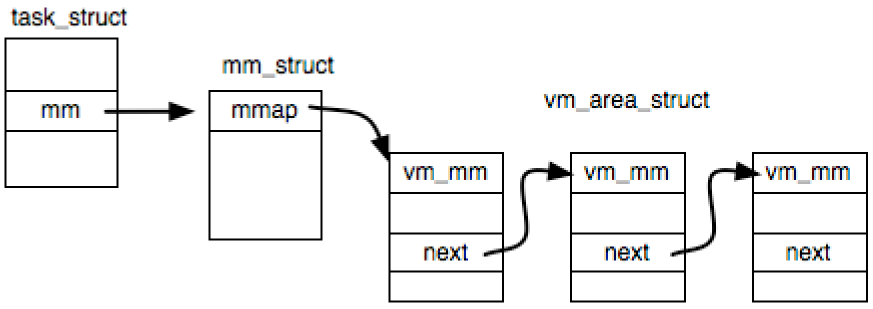
Details

Implement a system call int readAddr(void *p). Given a user space virtual address p, return the information about this address.

1- Is p already allocated/valid or not?
2- If yes, what is the start and end addresses of the virtual memory area containing p.
3- Is that virtual memory area readable, writable, and/or executable?
4- As shown below, you should iterate through every address in the user address space at an interval of PAGE_SIZE * 1024. Finally, print the number of valid/invalid addresses.

```
char *p = 0;
unsigned long validAddr = 0, invalidAddr = 0;
for( ; (unsigned long)p < TASK_SIZE; p += PAGE_SIZE * 1024 ) {
       ...
       int r = readAddr(p);
       print the information about p;
       ...
}
printf("%lu out of %lu addresses are valid\n", validAddr, validAddr +
invalidAddr);
```

You should make use of the mm field in task_struct. Specifically, mm is of type mm_struct, which contains a field mmap pointing to a list of nodes, each describing a virtual memory area (VMA) vm_area_struct. A VMA describes a range of virtual address space and the allowed access operations (VM_READ, VM_WRITE, VM_EXEC, etc.) in user space.



Tips

To obtain PAGE_SIZE and TASK_SIZE, please refer to the code below

```
#include <unistd.h>
unsigned long PAGE_SIZE = 0, TASK_SIZE = 0;

PAGE_SIZE = sysconf(_SC_PAGESIZE);
if(sizeof (void*) == sizeof (int)) // 32-bit system
       TASK_SIZE = 0xc0000000UL;
else // 64-bit system
       TASK_SIZE = (1UL << 47) - PAGE_SIZE;
```

Submission

Your submission should include the code (the kernel code modification should be submitted as a kernel patch), a readme file describing your design, how to compile / use your code and the contribution in the case of group programming, and a report which consists of the following parts:

- What are the APIs used to allocate memory in Linux user space, and when to use which.
- When an address p is dereferenced, you may encounter SIGSEGV. Describe how the system recognizes p is an invalid address and triggers a SIGSEGV. Hint: your answer should involve TLB, page table, and VMA.
- What information is saved at /proc/$pid/maps?