References

- http://gauss.ececs.uc.edu/Courses/e4022/code/memory/understanding.pdf
- Linux Kernel Programming - Book by Kaiwan N Billimoria
- https://github.com/davidhcefx/Translate-Virtual-Address-To-Physical-Address-in-Linux-Kernel
- https://www.linuxtopia.org/online_books/Linux_Kernel_Module_Programming_Guide/x323.html

LKM 1

How to run lkm1.c and sample output?

```
make clean; make

ps ax -eo pid,stat | grep -E 'R|R\+'
# output
2644 Rsl
29576 R+
29584 R+
30171 R+

sudo insmod lkm1.ko ; sudo rmmod lkm1 ; dmesg | tail -8
# output
[10357.328309] Module lkm1 inserted...
[10357.328313] List of running and runnable processes:
[10357.328374] TGID: 5399 | PID: 5399 | Name: code
[10357.328387] TGID: 21826 | PID: 21826 | Name: code
[10357.328398] TGID: 29576 | PID: 29576 | Name: python3
[10357.328399] TGID: 29584 | PID: 29584 | Name: python3
[10357.328401] TGID: 30203 | PID: 30203 | Name: insmod
[10357.328402] Total # of running and runnable processes: 5
[10357.344930] Exited from module lkm1

ps ax -eo pid,stat | grep -E 'R|R\+'
# output
29576 R+
29584 R+
30343 R+
30344 R+
```

ps vs output of lkm1

- The state field of the process descriptor describes what is currently happening to the process. It consists of an array of flags, each of which describes a possible process state. In the current Linux version, these states are mutually exclusive, and hence exactly one flag of state always is set; the remaining flags are cleared.
  - TASK_RUNNING:- The process is either executing on a CPU or waiting to be executed.
- **ps stands for process status**. It reports a snapshot of current processes. **Running /Runnable(R)**: Running process is currently the process that is being executed on the CPU. The runnable process is waiting in the queue and is in the ready state, meaning it has all it needs to be run and is waiting for a CPU core to execute it. The state corresponds to the running stage in the process life cycle.

NOTE: Both are giving almost same results with some variation. **ps** and **insmod** results are taken on different time so there is some difference. Two cpu intensive python program are running in background. So, both are giving pid = 29576 and 29584 as running process. **insmod** was active only when the module was inserted.

LKM 2

Assumption: heap calculation using (**end - start_brk**) and print only one process with maximum heap size.

How to run lkm2.c and sample output?

```
make clean; make
sudo insmod lkm2.ko ; sudo rmmod lkm2 ; sudo dmesg | tail -3
# output
[12064.560022] Loaded module lkm2...
[12064.560132] TGID: 2644 | PID: 2644 | Name: gnome-shell has the maximum heap memory: 137515008 bytes
[12064.583206] Exited from module lkm2
```

I iterate over all the process using for_each_process macros and then calculated heap using (**end - start_brk**) and stored the pid which has max heap.

LKM 3

Assumption: As discussed with sir, void *stack was expected.

How to run lkm3.c and sample output?

```
make clean; make
sudo insmod lkm3.ko ; sudo rmmod lkm3 ; sudo dmesg | tail -3
# output
[12397.345336] Module lkm3 loaded...
[12397.345343] TGID: 1 | PID: 1 | Name: systemd | Kernel Stack Pointer: 18446648331373019136 (0xffffa8ec400a8000)
[12397.362039] Exited from module lkm3
```

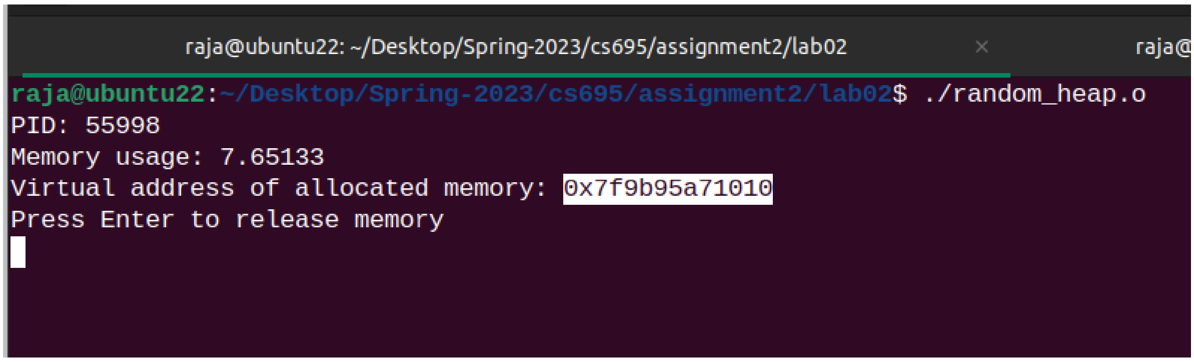
Find task_struct using **pid_task(find_vpid(1), PIDTYPE_PID)**. Then printing is straightforward.

LKM 4

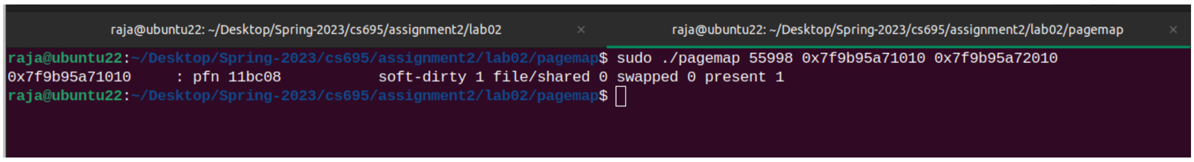
How to run lkm4.c and sample output?

```
make clean; make
# wrong pid check
# usage - sudo insmod lkm4.ko pid=<pid> virt_addr=<virtual_address>
sudo insmod lkm4.ko pid=45789 virt_addr=0xffff800000000045 ; sudo rmmod lkm4 ; sudo dmesg | tail -3
# output
[13088.864237] Module lkm4 loaded...
[13088.864241] Error: PID 45789 does not exist
[13088.878539] Exited from module lkm4
```

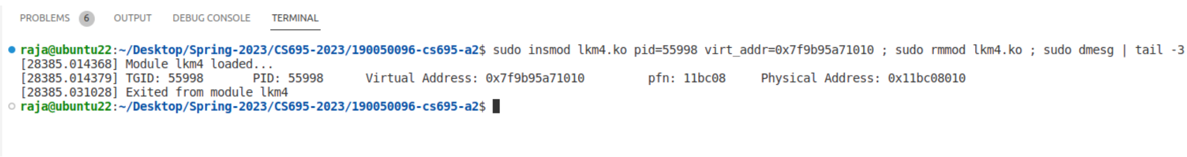
```
# address not present check
sudo insmod lkm4.ko pid=37958 virt_addr=0xffff600000000045 ; sudo rmmod lkm4 ;
sudo dmesg | tail -3
# output
[13194.296157] Module lkm4 loaded...
[13194.296161] Error - p4d: Virtual Address 0xffff600000000045 of PID 37958 is
not present
[13194.310422] Exited from module lkm4

------- Below results are generated on some other time ------ so different time
stamp ---------------------
```



```
raja@ubuntu22:~/Desktop/Spring-2023/cs695/assignment2/lab02$ ./random_heap.o
PID: 55998
Memory usage: 7.65133
Virtual address of allocated memory: 0x7f9b95a71010
Press Enter to release memory
```



```
raja@ubuntu22:~/Desktop/Spring-2023/cs695/assignment2/lab02/pagemap$ sudo ./pagemap 55998 0x7f9b95a71010 0x7f9b95a72010
0x7f9b95a71010     : pfn 11bc08          soft-dirty 1 file/shared 0 swapped 0 present 1
raja@ubuntu22:~/Desktop/Spring-2023/cs695/assignment2/lab02/pagemap$
```



```
raja@ubuntu22:~/Desktop/Spring-2023/CS695-2023/190050096-cs695-a2$ sudo insmod lkm4.ko pid=55998 virt_addr=0x7f9b95a71010 ; sudo rmmod lkm4.ko ; sudo dmesg | tail -3
[28385.014368] Module lkm4 loaded...
[28385.014379] TGID: 55998      PID: 55998      Virtual Address: 0x7f9b95a71010         pfn: 11bc08     Physical Address: 0x11bc08010
[28385.031028] Exited from module lkm4
raja@ubuntu22:~/Desktop/Spring-2023/CS695-2023/190050096-cs695-a2$
```

Find task_struct using **pid_task(find_vpid(pid), PIDTYPE_PID)**. I have use module_param to take pid and virt_addr as command line argument. Then used standard page walk method (using pgd, p4d, pud, pmd and pte) to calculate physical address. More details can be found in the lkm4.c

I run a process random_heap.o in which I allocated heap and get virtual address and then I inserted lkm.ko

random_heap

```
#include <iostream>
#include <unistd.h>

int main() {
    // Allocate a large block of memory on the heap
    char *large_block = new char[1000000000];

    // Print the process ID and memory usage
    std::cout << "PID: " << getpid() << std::endl;
    std::cout << "Memory usage: " << sysconf(_SC_PAGESIZE) *
sysconf(_SC_PHYS_PAGES) / (1024.0 * 1024.0 * 1024.0) << std::endl;
    std::cout << "Virtual address of allocated memory: " << static_cast<void*>
(large_block) << std::endl;
```

```
    std::cout << "Press Enter to release memory" << std::endl;
    std::cin.ignore();

    // Release the memory
    delete[] large_block;
    return 0;
}
```

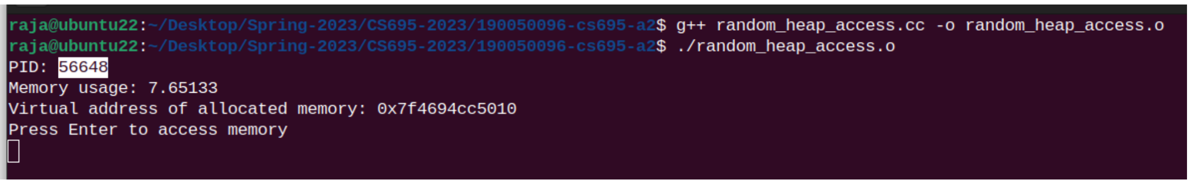
LKM 5

How to run lkm5.c and sample output?

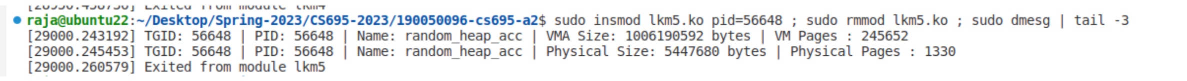
```
make clean; make
# I run a heap allocator
./random_heap.o
PID: 42236
Memory usage: 7.65133
Virtual address of allocated memory: 0x7f3075232010
Press Enter to access memory

# then lkm5
sudo insmod lkm5.ko pid=42236 ; sudo rmmod lkm5 ; sudo dmesg | tail -3
[14769.068520] TGID: 42236 | PID: 42236 | Name: random_heap.o | VMA Size:
1006194688 bytes | VM Pages : 245653
[14769.070388] TGID: 42236 | PID: 42236 | Name: random_heap.o | Physical Size:
5730304 bytes | Physical Pages : 1399
[14769.089861] Exited from module lkm5
```

Find task_struct using **pid_task(find_vpid(pid), PIDTYPE_PID)**. I have use module_param to take pid as command line argument. Then used vma to calculate size and walk page to check if physical page exists and calculated size accordingly. More details can be found in the lkm5.c

I run a process random_heap_access.o in which I allocated heap and then I inserted lkm5.ko.

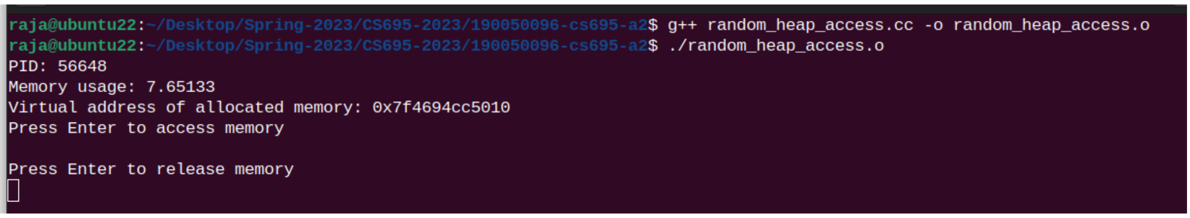


```
raja@ubuntu22:~/Desktop/Spring-2023/CS695-2023/190050096-cs695-a2$ g++ random_heap_access.cc -o random_heap_access.o
raja@ubuntu22:~/Desktop/Spring-2023/CS695-2023/190050096-cs695-a2$ ./random_heap_access.o
PID: 56648
Memory usage: 7.65133
Virtual address of allocated memory: 0x7f4694cc5010
Press Enter to access memory
```

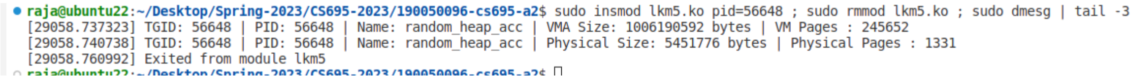


```
raja@ubuntu22:~/Desktop/Spring-2023/CS695-2023/190050096-cs695-a2$ sudo insmod lkm5.ko pid=56648 ; sudo rmmod lkm5.ko ; sudo dmesg | tail -3
[29000.243192] TGID: 56648 | PID: 56648 | Name: random_heap_acc | VMA Size: 1006190592 bytes | VM Pages : 245652
[29000.245453] TGID: 56648 | PID: 56648 | Name: random_heap_acc | Physical Size: 5447680 bytes | Physical Pages : 1330
[29000.260579] Exited from module lkm5
```

After I clicked enter page was accessed and more physical pages found but vm pages remain same. This shows that Linux has Lazy allocation.



```
raja@ubuntu22:~/Desktop/Spring-2023/CS695-2023/190050096-cs695-a2$ g++ random_heap_access.cc -o random_heap_access.o
raja@ubuntu22:~/Desktop/Spring-2023/CS695-2023/190050096-cs695-a2$ ./random_heap_access.o
PID: 56648
Memory usage: 7.65133
Virtual address of allocated memory: 0x7f4694cc5010
Press Enter to access memory

Press Enter to release memory
```



```
● raja@ubuntu22:~/Desktop/Spring-2023/CS695-2023/190050096-cs695-a2$ sudo insmod lkm5.ko pid=56648 ; sudo rmmod lkm5.ko ; sudo dmesg | tail -3
  [29058.737323] TGID: 56648 | PID: 56648 | Name: random_heap_acc | VMA Size: 1006190592 bytes | VM Pages : 245652
  [29058.740738] TGID: 56648 | PID: 56648 | Name: random_heap_acc | Physical Size: 5451776 bytes | Physical Pages : 1331
  [29058.760992] Exited from module lkm5
○ raja@ubuntu22:~/Desktop/Spring-2023/CS695-2023/190050096-cs695-a2$ □
```