Navid Malek
navidmalekedu@gmail.com
Project Phase 2

Operating Systems
Spring 2017

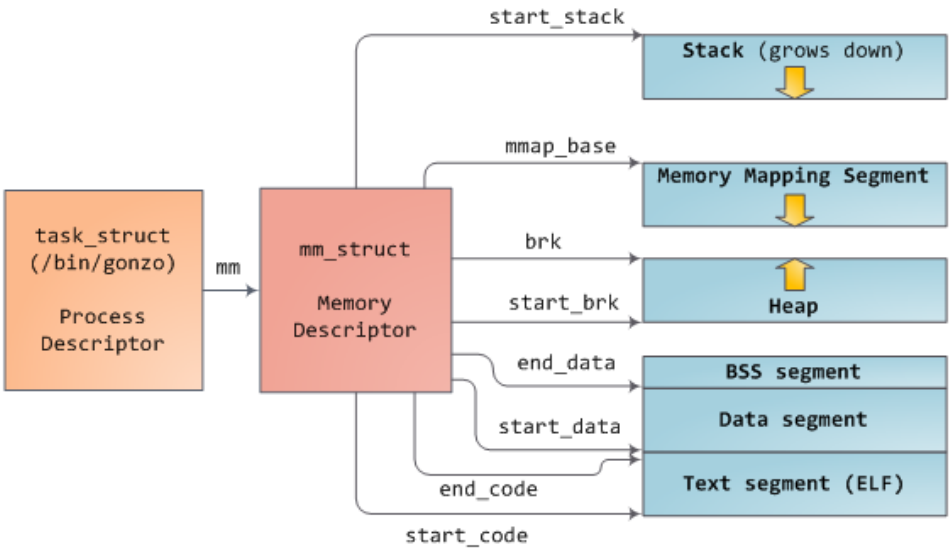
Part 2

Finding specified information and necessary functions to implement desired module

Goal:

- ✗ find start and end of code segment in memory for a process
- ✗ find start and end of data segment in memory for a process
- ✗ find BSS virtual address for process
- ✗ find entry point of a process

below there is a diagram that how mm_struct in task_struct manages memory segments of a process:



in task_struct there are two mm_struct's:

```
struct mm_struct        *mm;
struct mm_struct        *active_mm;
```

Navid Malek
navidmalekedu@gmail.com
Project Phase 2

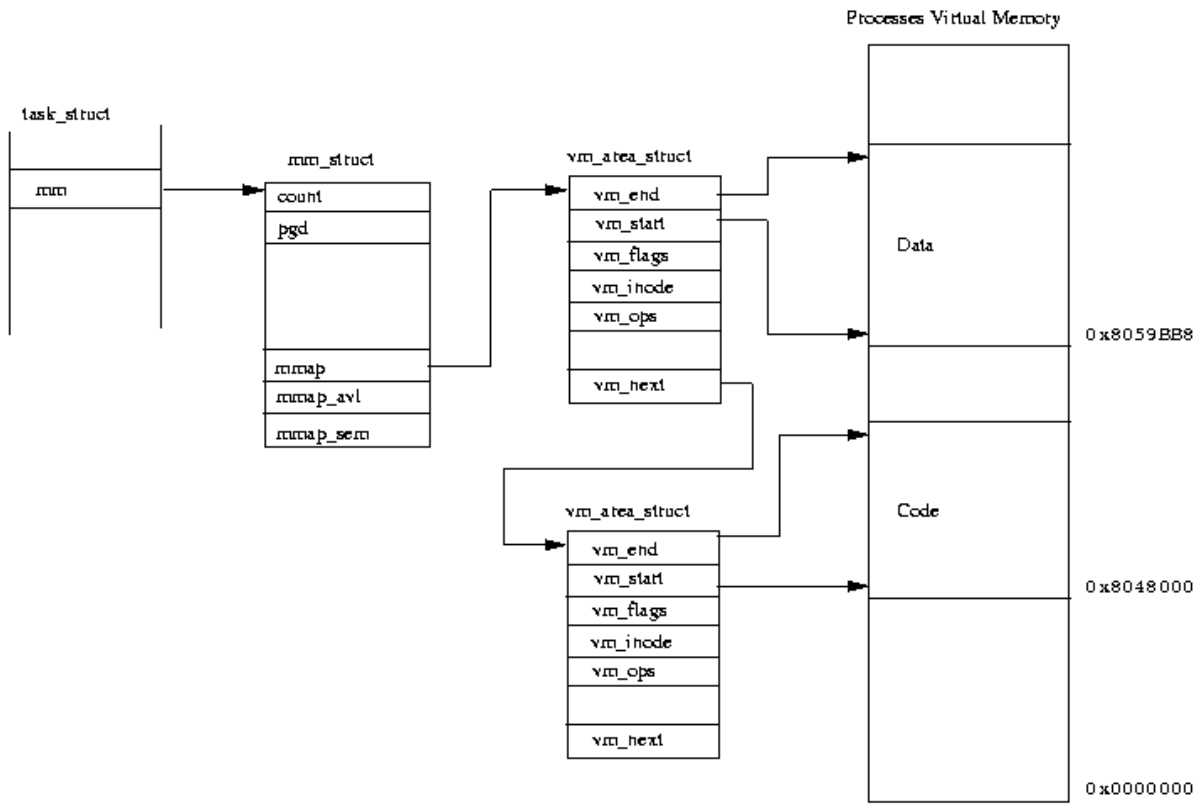
Operating Systems
Spring 2017

we will use *active_mm instead of *mm.

so till now :

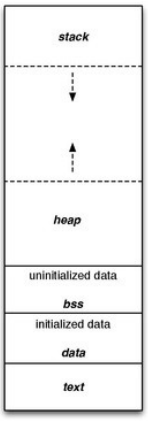
- ✔ find start and end of code segment
- ✔ find start and end of data segment

for the BSS vm address we have the vm_area_struct shown below:



according to http://linuxgazette.net/112/krishnakumar.html bss area is located in third VM strruct:

- ✔ find BSS virtual address

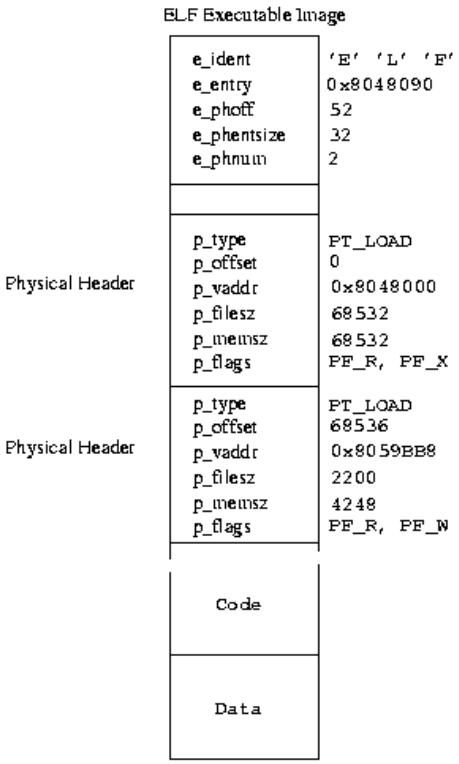


BUT also bss_start address is after data_segment end address, here I will only use third VM struct mentioned above.

Navid Malek
navidmalekedu@gmail.com
Project Phase 2

Operating Systems
Spring 2017

bellow there is diagram of ELF executable format:



e_entry is the entry point of the program(the value is address as shown in the diagram).

The entry point for the image, the first instruction for the program, is not at the start of the image but at virtual address *0x8048090* (`e_entry`)

- ✔ find Entry Point address

Note! saved_auxv[19] also points to entry point address

```
unsigned long start_brk, brk, start_stack;
unsigned long arg_start, arg_end, env_start, env_end;

unsigned long saved_auxv[AT_VECTOR_SIZE]; /* for /proc/PID/auxv */
```