10/5/22 Processes

Definition : Running instance of a program

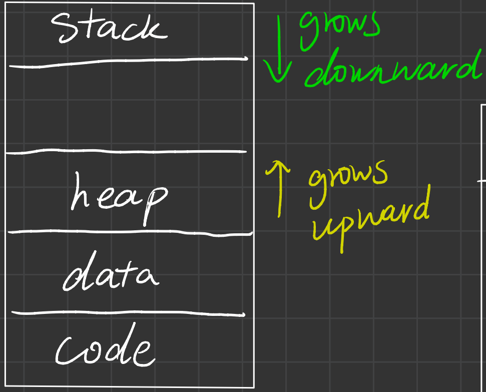
→ process is a unit of execution & scheduling (single threaded proc)

→ consists of ① Address Space ② CPU states ③ OS resources

→ unit of isolation (failure)

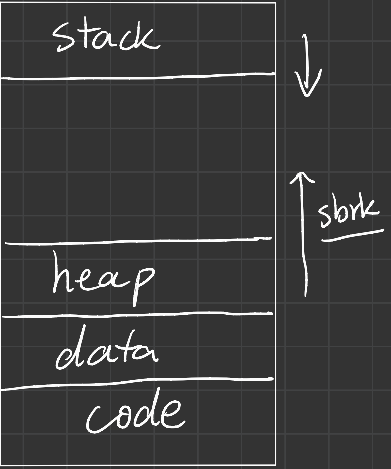
Process Implementation

- Address Space

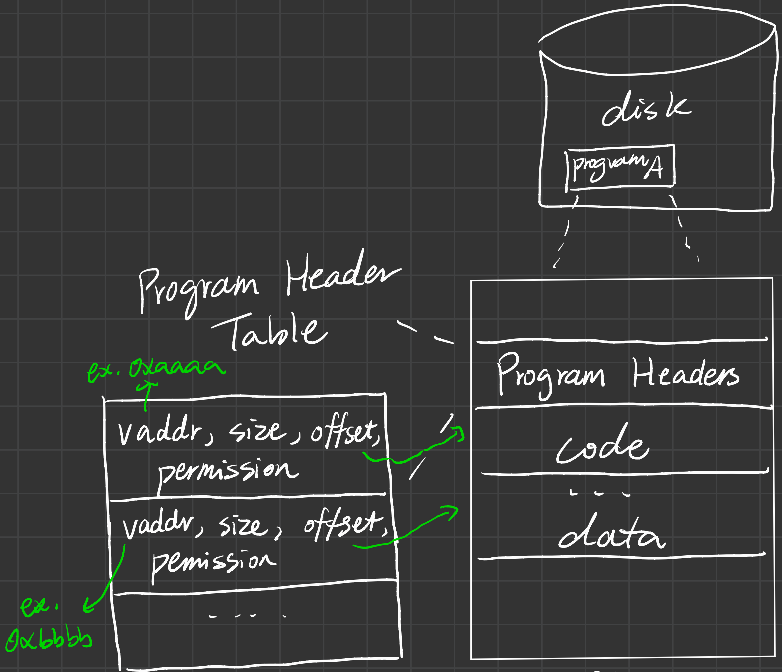


generic view of process memory

How did we get here?



Virtual Address Space for Process A



- vaddr = where to load the segment
- size = how many bytes
- offset = where the segment resides in the ELF file
- pemission = read only? read write?

ELF file (Executable file for program A)

In addition to the address space, kernel also needs to set up arguments for new processes.

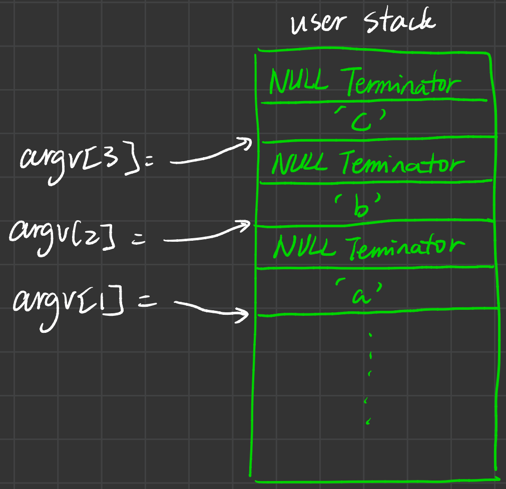
int main (int argc, char** argv)

→ assume we have argc = 4, argv = ["foo", "a", "b", "c"]

register rdi = 4 (calling convention)

register rsi = ? where is the string array? Where are the strings?

☆ user stack



[string args & argv array are pushed onto the stack]

CPU states : pc = elf.entry
sp = top of stack
registers = rdi, rsi populated.

OS Management

→ vspace (xk)

→ per process, tracks VAS of a process

(random addresses for examples)

Example: vregion [heap] 0xabcde — 0xbbcde

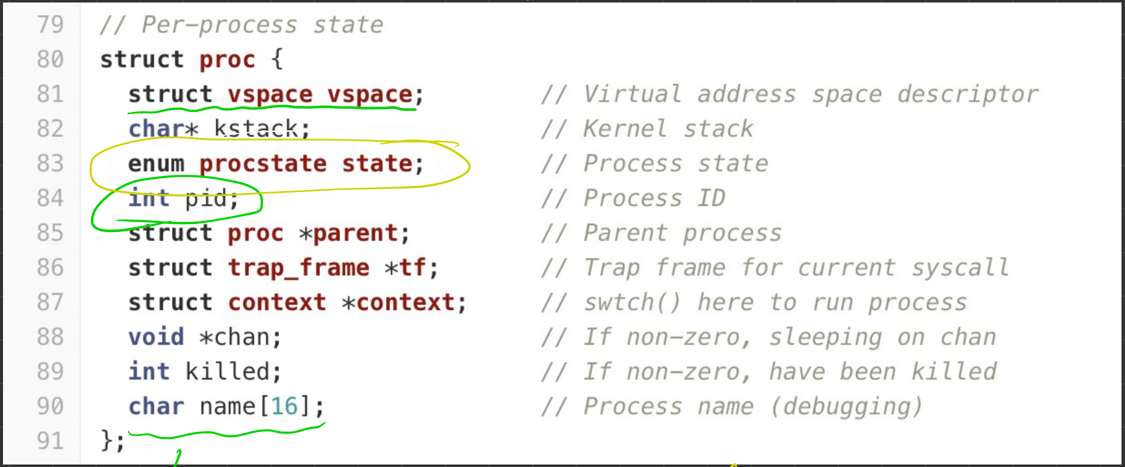
vregion [stack] 0xfe000 — 0xff000

vregion [code] - - - - - -

vregion [data] - - - - - -

Process Control Block

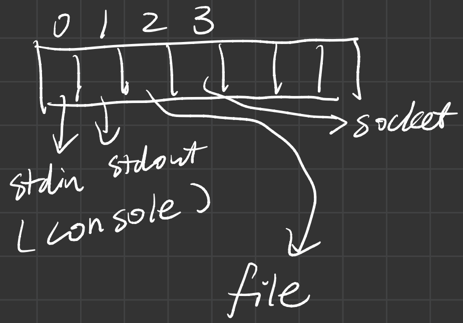
inc/proc.h



```
// Per-process state
struct proc {
  struct vspace vspace;        // Virtual address space descriptor
  char* kstack;                // Kernel stack
  enum procstate state;        // Process state
  int pid;                     // Process ID
  struct proc *parent;         // Parent process
  struct trap_frame *tf;       // Trap frame for current syscall
  struct context *context;     // swtch() here to run process
  void *chan;                  // If non-zero, sleeping on chan
  int killed;                  // If non-zero, have been killed
  char name[16];               // Process name (debugging)
};
```

processes can have the same name

File Descriptor Table



Additional fields:

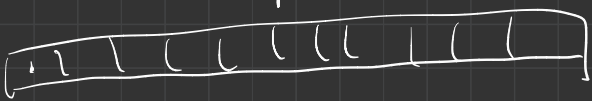
→ scheduling priority

→ file descriptor table (lab 1)

PCB → Linux: kernel heap

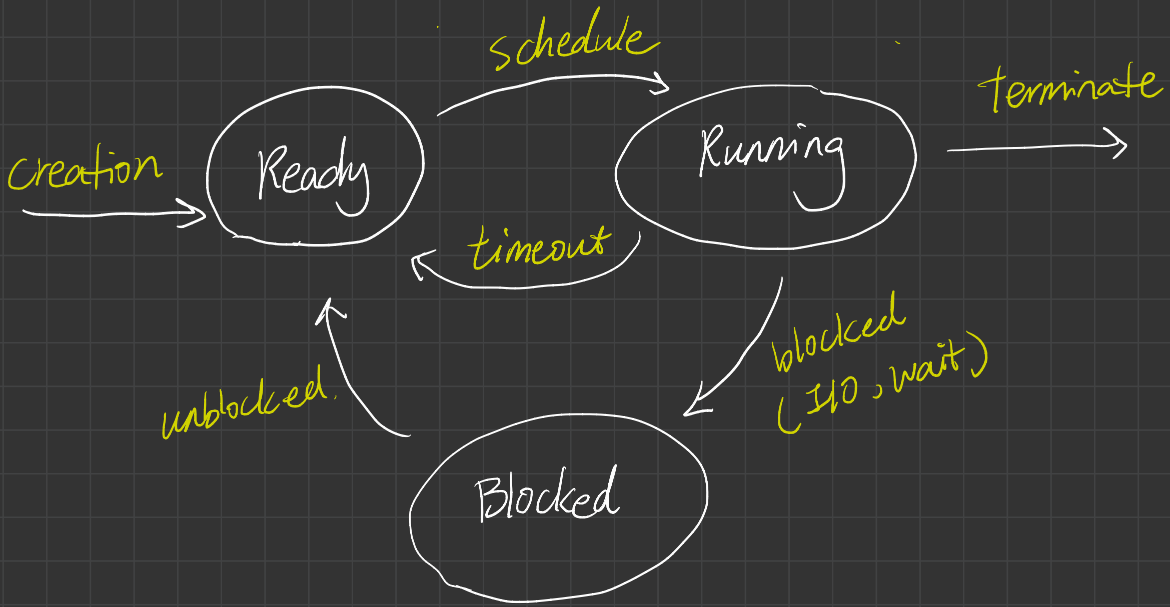
Allocation:

→ xk:



process table

Process Life Cycle



Syscall Validation.

user addresses.

byte

int, pointers (buffer, strings)

vspace

read(fd)