Question 1

1.

Struct proc = pcb of xv6

```
struct proc {
  uint sz;                     // Size of process memory (bytes)
  pde_t* pgdir;                // Page table
  char *kstack;                // Bottom of kernel stack for this process
  enum procstate state;        // Process state
  int pid;                     // Process ID
  struct proc *parent;         // Parent process
  struct trapframe *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
  struct file *ofile[NOFILE];  // Open files
  struct inode *cwd;           // Current directory
  char name[16];               // Process name (debugging)
}
```

```
struct context {
  uint edi;
  uint esi;
  uint ebx;
```

```
  uint ebp;
  uint eip;
};

enum procstate { UNUSED, EMBRYO, SLEEPING,
RUNNABLE, RUNNING, ZOMBIE };

// Per-process state
```

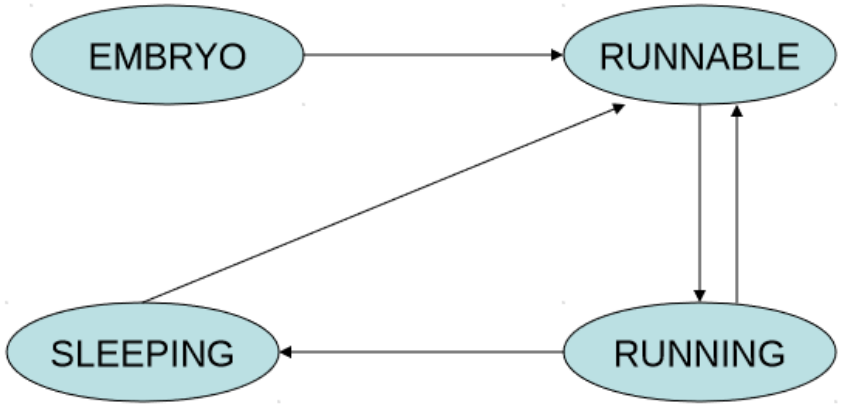
2.

- **Sz** => size of process memory(bytes)

  برای دانستن سایز مموری فرآیند به بایت از آن استفاده میشود.

- **State** => process state

  وضعیت فرآیند را مشخص میکند.



EMBRYO => The new process is currently being created

RUNNABLE => Ready to run

RUNNING => Currently executing

SLEEPING => Blocked for an I/O

Other states ZOMBIE (later)

- **Context** =>

مقادیر رجیسترهایی که برای اجرا لازم هستند نگهداری میشود و برای فرایند های بعدی این مقادیر ازpcb روی سخت افزار load میشود.
با انجام Context swith به جایی که باید فرآیند اجرا شود switch میکند و مقادیر رجیستر های ذخیره شده را restoreمیکند.
رجیسترهایی مثل :
Status register, page-table orogin page table length

- **Ofile** => open files

فایل های باز را مشخص میکند.

- **Killed** => number of killed processes

تعداد فرآیندهای کشته شده را مشخص میکند.