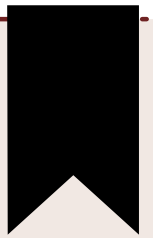
User Environment



- Environment <=> Process
- Implement protected user-mode environment
- Create single Envi → Load Image → Run
- GCC's Inline assembly language feature
- Context switch, Handling Traps
- Handle System Calls & Exceptions

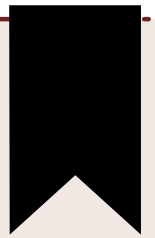
Comparison

- Struct env
- env_free_list
- NENV
- envs array
- struct Env *curenv = NULL

- Struct pageinfo
- page_free_list
- npages
- pages array

```
struct Env {
        struct Trapframe env_tf;

        struct Env *env_link;

        envid_t env_id;

        envid_t env_parent_id;

        enum EnvType env_type;

        unsigned env_status;

        uint32_t env_runs;

        pde_t *env_pgdir;};
```

Environment Array

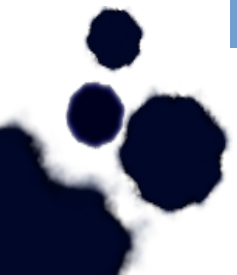


• Save register context when environment is not about to run.

•On termination, the kernel may re-allocate the same Env array.

• The new envs will have a different env_id, even though re-using the same slot .

0	Uniquifier <21 bits>	Eid <10bits>

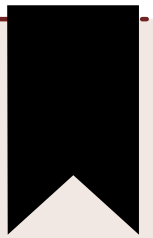


- In mem_init() allocate the envs array & map it to UENVS region in memory as read-only.

```
*    ULIM      ------> +------------------------------+ 0xef800000
*                     |  Cur. Page Table (User R-)   | R-/R-  PTSIZE
*    UVPT       ----> +------------------------------+ 0xef400000
*                     |          RO PAGES            | R-/R-  PTSIZE
*    UPAGES     ----> +------------------------------+ 0xef000000
*                     |           RO ENVS            | R-/R-  PTSIZE
* UTOP,UENVS ------> +------------------------------+ 0xeec00000
* UXSTACKTOP -/       |     User Exception Stack     | RW/RW  PGSIZE
*                     +------------------------------+ 0xeebff000
```

- Initialize envs, segment descriptors & GDT
- boot_map_region
- boot_alloc
- env_init

Setting the VM



- Set the segment Descriptor(assembly code)
- env_init_percpu
- pde_t *env_pgdir; // Kernel VA address of PD
- Allocate a page for PD. Each environment has a Page Table.
- env_pgdir array is mapped to array of PD entries.
- env_setup_vm

Setting the Env to run

- Allocate memory & init env (env_alloc)
- Initialize parent_id, type to default value.
- Load ELF image to user Addr Space (load_icode)
- Phy memory allocation for env.
- Map it to VA
- Function : region_alloc
- Allocate Page, Insert
- i386_init → env_create → load_icode →region_alloc