

# **OPERATING SYSTEMS (THEORY)**

## **LECTURE - 2**

**K.ARIVUSELVAN**

*Assistant Professor (Senior) – (SITE)*

*VIT University*

# PROCESS MANAGEMENT

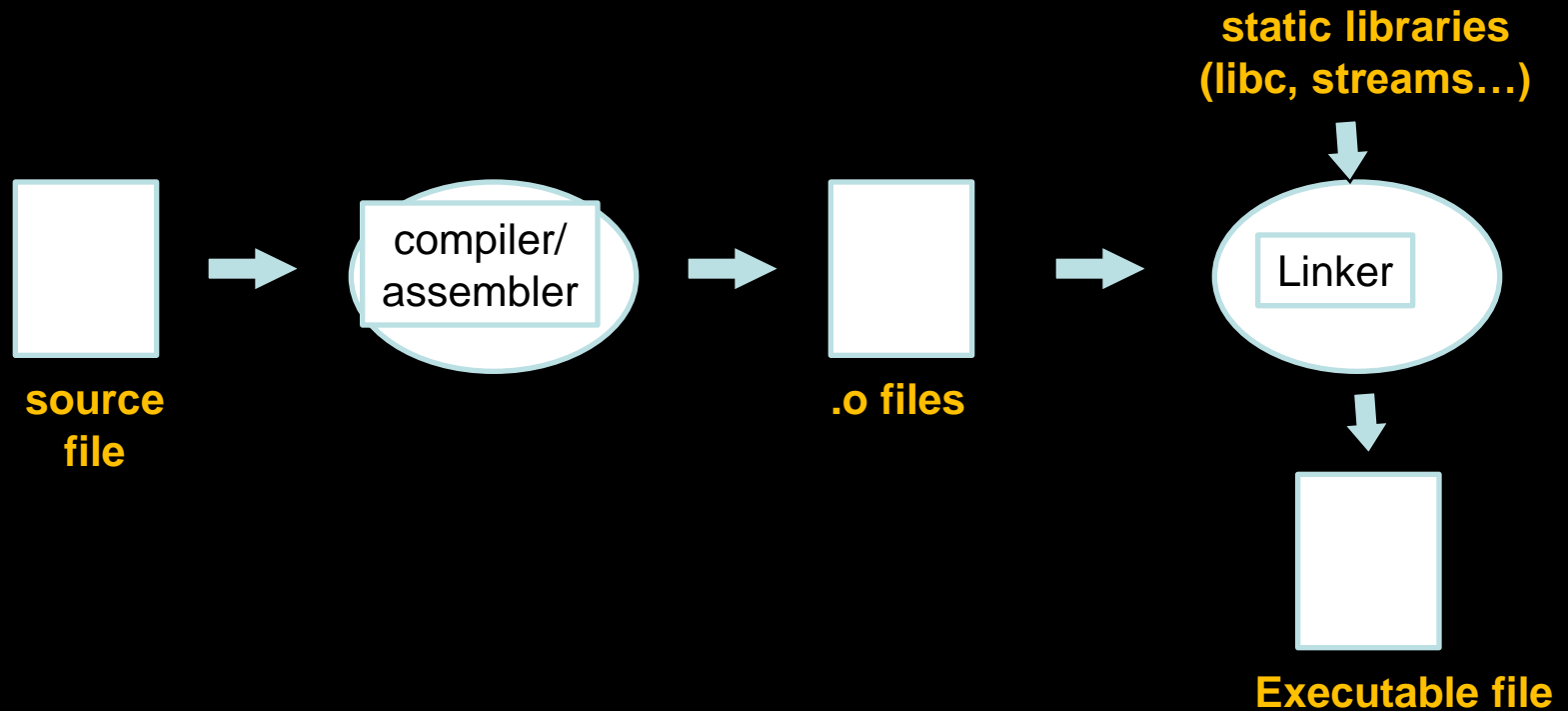
# Program

A program consists of:

**Code:** machine instructions

**Data:** variables stored and manipulated in memory

# Preparing a Program



# PROCESS

- Program in Execution

You might think of it as...

- The **collection of data structures** that fully describes **how far the execution** of the program has **progressed**.

# PROCESS

**A Process includes 4 Segments:**

**CODE / TEXT:** Holds program

**DATA:** Holds program variables

**HEAP:** Holds intermediate computation data generated during run time

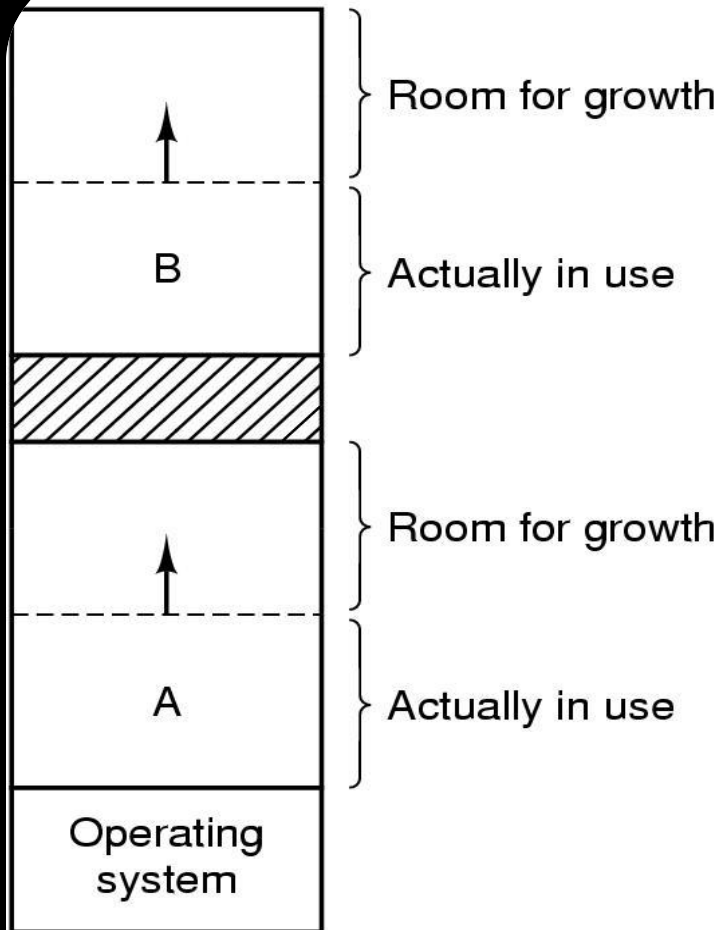
**STACK:** Holds

=> Local variables

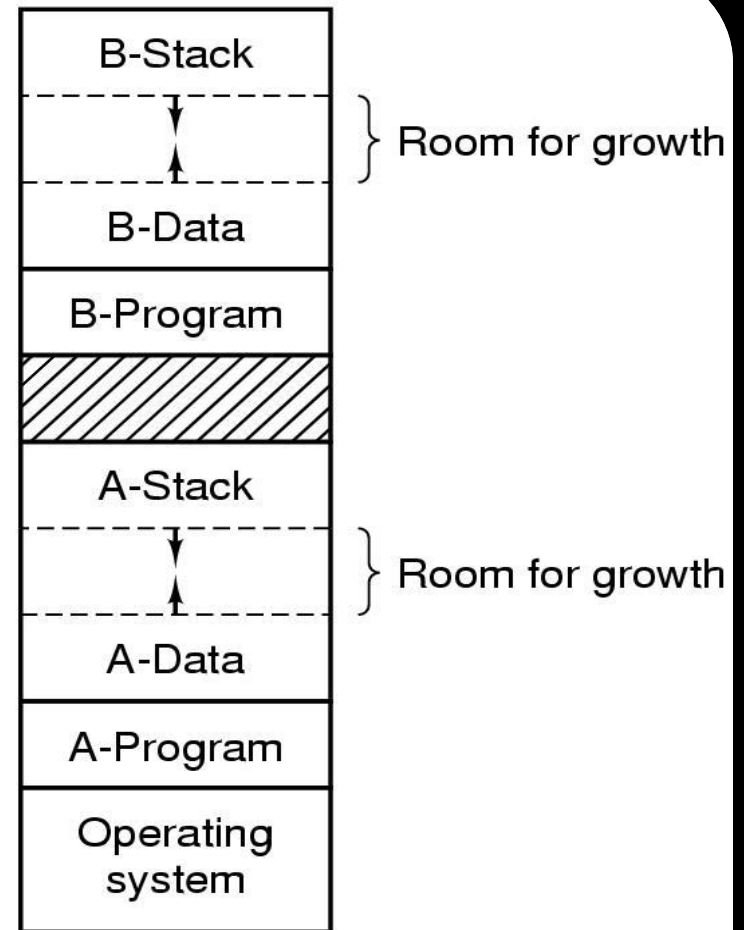
=> Temporaries & Procedure Calls

=> Return addresses

# Processes in Memory



(a)

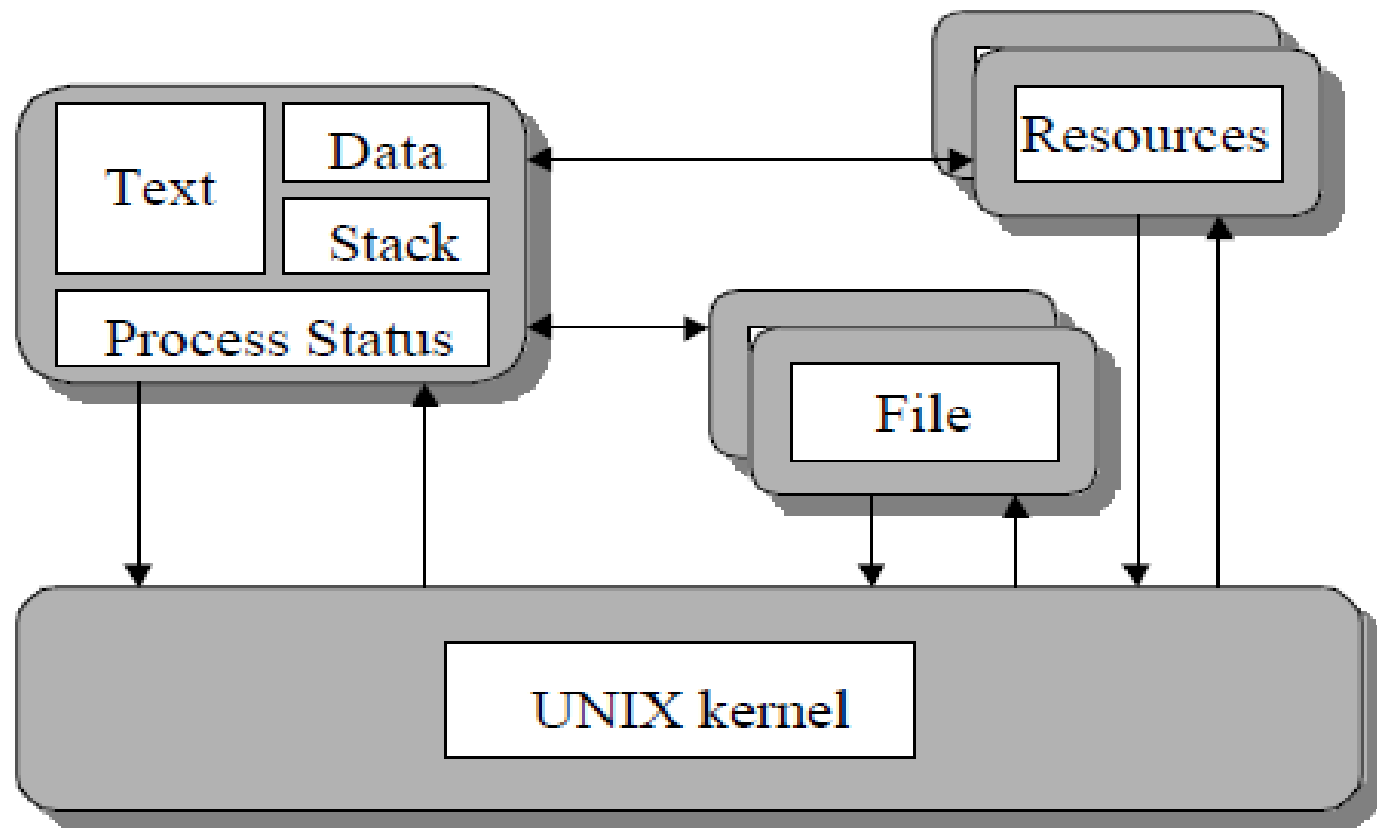


(b)

(g)

(p)

# A Unix Process





## PROGRAM

Instructions in **any** programming language

### Passive Entity

{Content of file stored on disk}

Resides in the **Secondary** storage

## PROCESS

Instruction execution in **machine** code

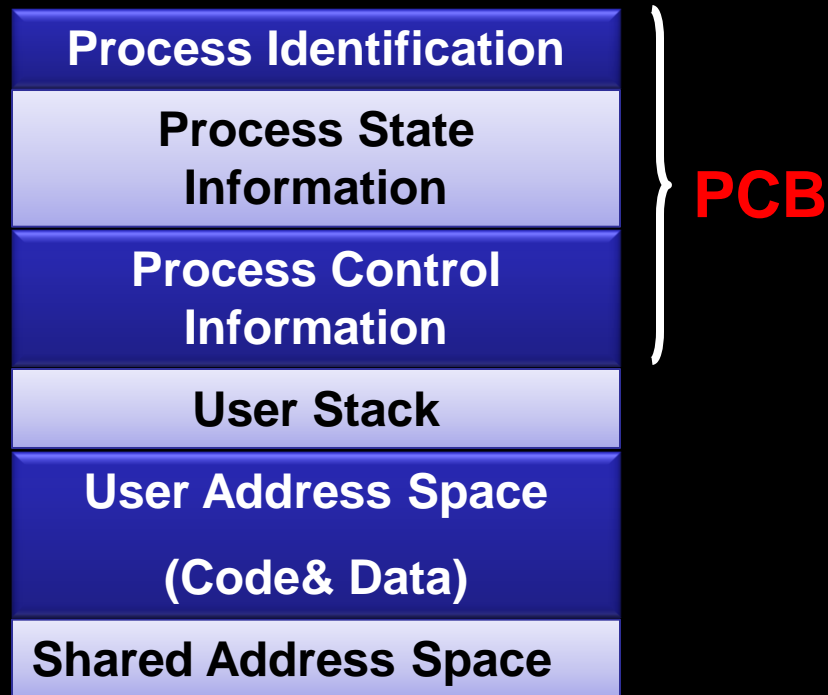
### Active Entity

{Contains program counter, specifying next instruction to execute}

Resides in **Main** memory

# PROCESS IMAGE

- Collection of CODE, DATA, STACK & ATTRIBUTES

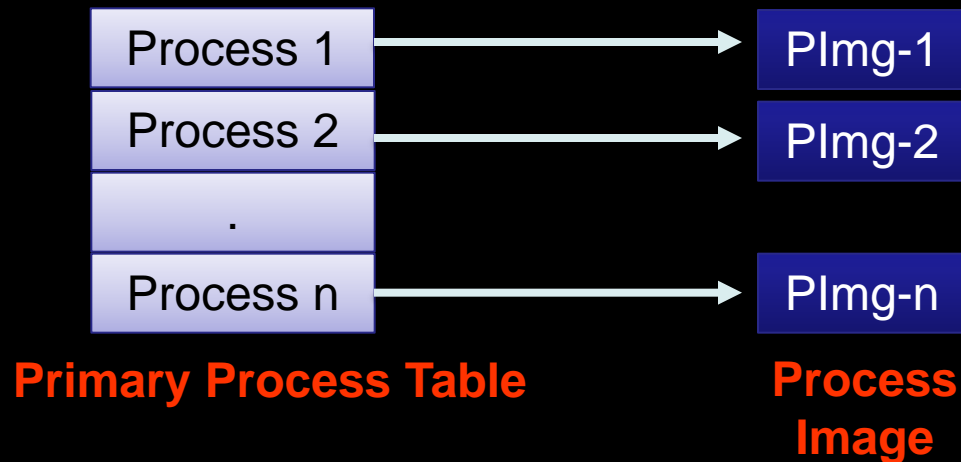


- The Collection of attributes is called as **Process Control Block (PCB)** or **Task Control Block (TCB)**

- The Process image is stored on the disk as **contiguous block**, this block may be:

- => Fixed Size (Pages)

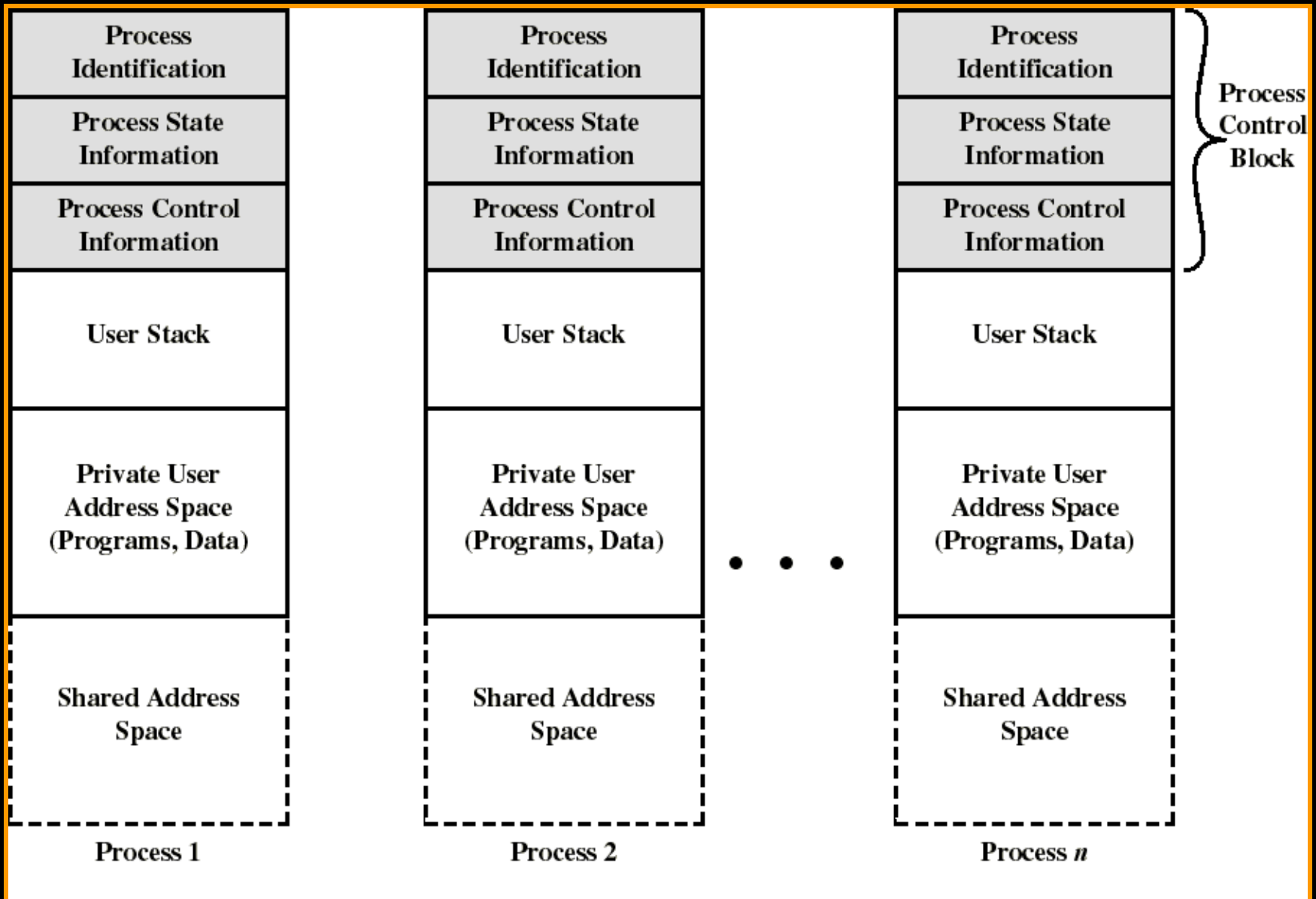
- => Variable Size (Segments)



- The address of each Process image is stored on **Primary Process Table**

- \* To maintain the process, its image must be brought into main memory

# Process Images in Main Memory



# Structure of PCB

**Unique ID** provided by OS on process creation

New, Ready, Running, Suspend, Terminated

May be **Internal** (Set by OS) or **External** (depends on external factors)

Includes Pointer to the

=> Program **code**

=> **Data** associated with this process

=> **Memory block** shared with other process

Includes

=> I/O **requests**, I/O **devices** assigned to this process

=> **List of Files** in use by the process

Identifier

State

Priority

Program Counter

**Address** of next instruction to be executed

Memory Pointer

Context Data

**Data** present in **Registers**, while process is **executing**

I/O Status Information

Accounting Information

Includes the amount

=> Amount of **processor time** used

=> Time limits`

# Linux PCB Structure (task\_struct)

```
struct task_struct {
volatile long state; Execution state
unsigned long flags;
int sigpending;
mm_segment_t addr_limit;
struct exec_domain *exec_domain;
volatile long need_resched;
unsigned long ptrace;
int lock_depth;
unsigned int cpu;
int prio, static_prio;
struct list_head run_list;
prio_array_t *array;
unsigned long sleep_avg;
unsigned long last_run;
unsigned long policy;
unsigned long cpus_allowed;
unsigned int time_slice, first_time_slice;
atomic_t usage;
struct list_head tasks;
struct list_head ptrace_children;
struct list_head ptrace_list;
struct mm_struct *mm, *active_mm; Memory mgmt info
struct linux_binfmt *binfmt;
int exit_code, exit_signal;
int pdeath_signal;
unsigned long personality;
int did_exec:1;
unsigned task_dumpable:1;
pid_t pid; Process ID
pid_t pgrp;
pid_t tty_old_pgrp;
pid_t session;
pid_t tgid;
int leader;
struct task_struct *real_parent;
struct task_struct *parent;
struct list_head children;
struct list_head sibling;
struct task_struct *group_leader;
struct pid_link pids[PIDTYPE_MAX];
wait_queue_head_t wait_chldexit;
struct completion *vfork_done;
int *set_child_tid;
int *clear_child_tid; Priority
unsigned long rt_priority;
```

```
unsigned long it_real_value, it_prof_value, it_virt_value;
unsigned long it_real_incr, it_prof_incr, it_virt_incr;
struct timer_list real_timer;
struct tms times; Accounting info
struct tms group_times;
unsigned long start_time;
long per_cpu_utime[NR_CPUS], per_cpu_stime[NR_CPUS];
unsigned long min_flt, maj_flt, nswap, cmin_flt, cmaj_flt,
cnsnap;
int swappable:1;
uid_t uid, euid, suid, fsuid; User ID
gid_t gid, egid, sgid, fsgid;
int ngroups;
gid_t groups[NGROUPS];
kernel_cap_t cap_effective, cap_inheritable, cap_permitted;
int keep_capabilities:1;
struct user_struct *user;
struct rlimit rlim[RLIM_NLIMITS];
unsigned short used_math;
char comm[16];
int link_count, total_link_count;
struct tty_struct *tty;
unsigned int locks;
struct sem_undo *semundo;
struct sem_queue *sem_sleeping; CPU state
struct thread_struct thread;
struct fs_struct *fs;
struct files_struct *files; Open files
struct namespace *namespace;
struct signal_struct *signal;
struct sighand_struct *sighand;
sigset_t blocked, real_blocked;
struct sigpending pending;
unsigned long sas_ss_sp;
size_t sas_ss_size;
int (*notifier)(void *priv);
void *notifier_data;
sigset_t *notifier_mask;
void *tux_info;
void (*tux_exit)(void);
u32 parent_exec_id;
u32 self_exec_id;
spinlock_t alloc_lock;
spinlock_t switch_lock;
void *journal_info;
unsigned long ptrace_message;
siginfo_t *last_siginfo;
};
```