

浙大城市学院实验报告

课程名称 操作系统原理实验

实验项目名称 实验二 认识PCB

学生姓名 司晨旭 专业班级 计算2002 学号 32001019

请查阅教材（表 3.4， 3.5 和 3.6）和资料，标出 Linux 的 PCB（task_struct）中标红字段的解释。

struct task_struct {
.....

volatile long	state; //进程状态
void	*stack; //进程内核栈
unsigned int	flags; //标记
unsigned int	ptrace; //系统调用
int	prio; //进程调度
int	static_prio;
int	normal_prio;
unsigned int	policy;
int	nr_cpus_allowed;
cpumask_t	cpus_allowed;
struct sched_info	sched_info; //用于调度器统计进程运行信息
struct list_head	tasks; //用于构架进程链表
struct mm_struct	*mm; // 进程所拥有的用户空间的内存描述符
struct mm_struct	*active_mm;
int	exit_state; //进程状态参数
int	exit_code;
int	exit_signal;
int	pdeath_signal; //父进程终止时会发送信号
unsigned	sched_reset_on_fork:1;
unsigned	sched_contributes_to_load:1;
unsigned	sched_migrated:1;
unsigned	sched_remote_wakeup:1;
unsigned long	atomic_flags; /* Flags requiring atomic access. */
struct restart_block	restart_block;
pid_t	pid; //进程pid
pid_t	tgid; // 线程组的领头线程的pid成员的值
struct task_struct__rcu	*real_parent; //父进程
struct task_struct__rcu	*parent;
struct list_head	children; //链表的头部，链表的所有元素都是它的子进
程	
struct list_head	sibling;
struct task_struct	*group_leader;
struct pid	*thread_pid; //进程标识号

```
char
文件名称，进程名
struct fs_struct
struct files_struct
struct signal_struct
struct thread_struct
};

comm[TASK_COMM_LEN]; //去除路径以后的可执行

*fs; //文件系统信息结构体
*files; //打开文件相关信息结构体
*signal; //进行信号处理
thread; //该进程在特点CPU下的状态
```