**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 comm[TASK\_COMM\_LEN];//去除路径以后的可执行文件名称，进程名**

**struct fs\_struct \*fs;//文件系统信息结构体**

**struct files\_struct \*files;//打开文件相关信息结构体**

**struct signal\_struct \*signal;//进行信号处理**

**struct thread\_struct thread;//该进程在特点CPU下的状态**

};