

# 进程基础 (六)

(一般要脱离父系、终端什么的！)  
具体创建，需的啥来这查就行

胎入出，终端不入出的程序，用于保护



守护进程

什么的...

主讲：大海老师

# 课程目标：

守护进程特点（了解）

会话、控制终端（了解）

创建守护进程（熟练）

小结

# 守护进程

守护进程(Daemon Process)是Linux三种进程类型之一

是 Linux 中的后台服务进程

是一个生存期较长的进程，通常独立于控制终端并且周期性地执行某种任务或等待处理某些发生的事件

# ✓ 守护进程特点

✓ 始终在后台运行

✓ 独立于任何终端

✓ 周期性的执行某种任务或等待处理特定事件

# 守护进程 - 相关概念

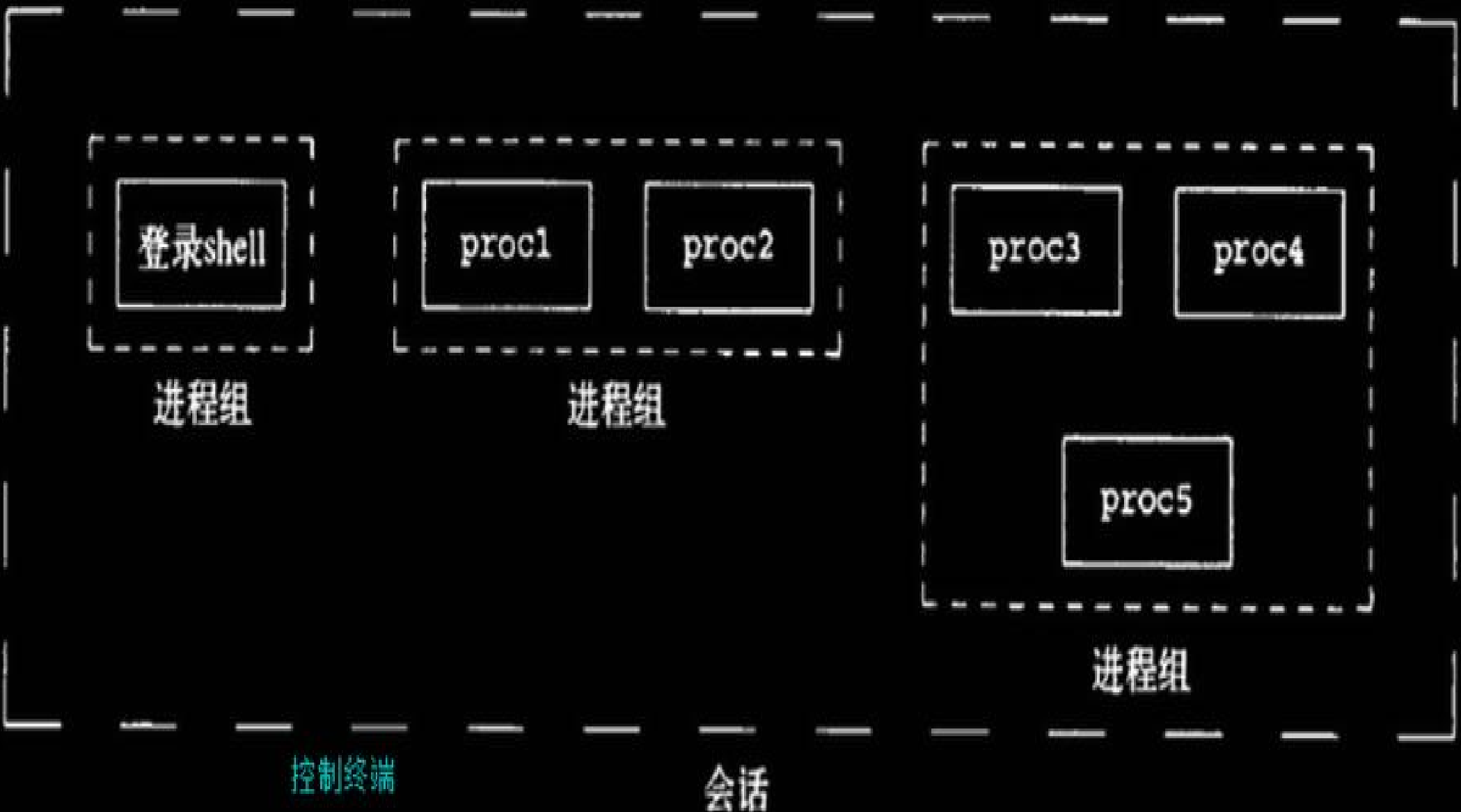
✓ 进程组 (Process Group) : 进程集合, 每个进程组有一个组长 (Leader), 其进程 ID 就是该进程组 ID。

(对应一个程序)

✓ 会话 (Session) : 进程组集合, 每个会话有一个组长, 其进程 ID 就是该会话组 ID。

(对应一个控制终端?)

✓ 控制终端 (Controlling Terminal) : 每个会话可以有一个单独的控制终端, 与控制终端连接的 Leader 就是控制进程 (Controlling Process)。



进程组和会话中的进程安排

# 守护进程创建(一)

- 创建子进程，父进程退出

```
if (fork() > 0) {  
    exit(0);  
}
```

- 子进程变成孤儿进程，被init进程收养
- 子进程在后台运行

# 守护进程创建(二)

- 子进程创建新会话

```
if (setsid() < 0) {  
    exit(-1);  
}
```

- 子进程成为新的会话组长

- 子进程脱离原先的终端



# ✓ 守护进程创建(三)

✓ 更改当前工作目录

```
chdir( "/" );  
chdir( "/tmp" );
```

✓ 守护进程一直在后台运行，其工作目录不能被卸载

✓ 重新设定当前工作目录cwd

# 守护进程创建(四)

- 重设文件权限掩码

```
if (umask(0) < 0) {  
    exit(-1);  
}
```

➤ 文件权限掩码设置为0

➤ 只影响当前进程

# 守护进程创建(五)

- 关闭打开的文件描述符

```
int i;  
for(i=0; i<3; i++) {  
    close(i);  
}
```

➤ 关闭所有从父进程继承的打开文件

➤ 已脱离终端，stdin / stdout / stderr无法再使用

Q 那以上也不全吧？ 嗯对，去父进程一看就知道了吧。

# 守护进程 – 示例1

创建守护进程，每隔1秒将系统时间写入文件time.log

```
int main() {  
    pid_t pid;  
    FILE *fp;  
    time_t t;  
    int i;  
  
    if ((pid = fork()) < 0) {  
        perror( "fork" ); exit(-1);  
    }  
    else if (pid > 0) {  
        exit(0);  
    }  
}
```

# 守护进程 – 示例2

```
setsid();
umask(0);
chdir( "/tmp" );
for (i=0; i< 3; i++) {
    close(i);
}
if ((fp = fopen( "time.log" , "a" )) == NULL) {
    perror( "fopen" ); exit(-1); }
while ( 1 ) {
    time(&t);
    fprintf(fp, "%s" , ctime(&t));
    fflush(fp);
    sleep(1);
}
```

# 进程小结

守护进程特点

守护进程创建步骤

