



北京邮电大学

Beijing University of Posts and Telecommunications

命令获取信息的方法

蒋砚军 北京邮电大学计算机学院



► Linux系统命令和用户程序(ap)运行时获取信息的方式



■ Linux系统命令和用户程序(ap)

- ◆ 从操作系统看，在地位上相同，都属于用户态程序
- ◆ 运行时需要获取的信息包括配置信息、处理方式（选项参数）、被处理的对象

■ 配置信息等硬编码是不可取的

硬编码需要编程时就确定服务器的地址，
程序运行时就无法改变，太不灵活

```
#define SERVER "180.249.151.131"
main()
{
    ...
}
```

■ 运行时获取信息的常见方式

易变性从小到大为

- ◆ 配置文件
- ◆ 环境变量
- ◆ 命令行参数
- ◆ 交互式键盘输入

► 运行时获取信息的方式



■ 配置文件

- ◆ 一般较复杂的程序会提供配置文件以存储配置信息或者偏好配置信息
- ◆ 分为**系统级偏好设置**和**用户级偏好设置**，例如bash的/etc/profile 和 ~/.bash_profile
- ◆ 配置文件提供了灵活性（同一个程序文件因用户不同读取的配置文件不同而表现不同），变更这些信息不很方便，**一般不需要变化的配置信息或选项信息存入配置文件**，持久化存储

■ 环境变量

命令env可以打印出当前的环境变量。

- ◆ 一般是些与“环境”相关的配置或选项信息，信息量不大。**这些选择在一段时间内反复使用同一个命令或者不同命令时保持不变**。例如：LANG(语言选择), HOME(主目录), TERM(终端类型), PATH(可执行文件的查找路径), CLASSPATH(类库查找路径), CVSROOT
- ◆ 虽然运行的程序（可执行文件）是完全相同的一个文件，程序通过获得环境变量感知环境的不同，控制自己的行为。
- 3 ◆ 环境变量值的获取与设置：C语言有库函数getenv(), 用户设置环境变量的方法也很简单
- ◆ 性能问题：比读取配置文件需要的系统开支要小

► 使用环境变量



```
#include <stdio.h>
#include <stdlib.h>

#define DEFAULT_SERVER "180.249.151.131"

int main(void)
{
    char *server;

    server = getenv("SERVER");
    if (server == NULL)
        server = DEFAULT_SERVER;

    printf("Server [%s]\n", server);
}
```

► 运行时获取信息的方式

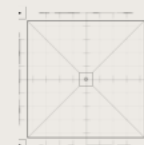


■ 命令行参数

- ◆ **程序启动之前**指定：通过命令行参数，操作员输入命令时提供处理选项和操作对象
- ◆ 每个命令都不同，命令运行完之后，对后续命令无影响
- ◆ 命令行参数的三种风格

■ 交互式键盘输入：这种方式在Linux命令中极少使用

程序启动之后通过计算机与操作员之间的人机交互获取信息，C语言scanf(), fgets()函数





北京邮电大学

Beijing University of Posts and Telecommunications



谢谢