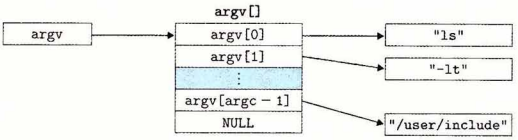

图 8-20　参数列表的组织结构

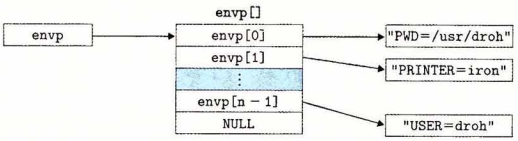

图 8-21　环境变量列表的组织结构

在 execve 加载了 filename 之后，它调用 7.9 节中描述的启动代码。启动代码设置栈，并将控制传递给新程序的主函数，该主函数有如下形式的原型

```
int main(int argc, char **argv, char **envp);
```

或者等价的

```
int main(int argc, char *argv[], char *envp[]);
```

当 main 开始执行时，用户栈的组织结构如图 8-22 所示。让我们从栈底(高地址)往栈顶(低地址)依次看一看。首先是参数和环境字符串。栈往上紧随其后的是以 null 结尾的指针数组，其中每个指针都指向栈中的一个环境变量字符串。全局变量 environ 指向这些指针中的第一个 envp[0]。紧随环境变量数组之后的是以 null 结尾的 argv[ ]数组，其中每个元素都指向栈中的一个参数字符串。在栈的顶部是系统启动函数 libc_start_main(见 7.9 节)的栈帧。

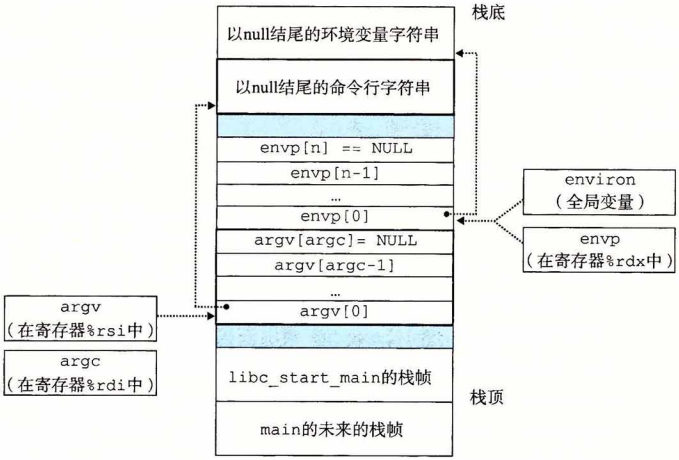

图 8-22　一个新程序开始时，用户栈的典型组织结构