

linux下的c编程学习笔记

郝俊禹*

Contents

1	socket网络编程	2
1.1	常见结构体	2
1.1.1	sockaddr	2
1.1.2	sockaddr_in	2
1.2	主要函数	2
1.2.1	socket	2
1.2.2	bind	3
1.2.3	connect	3
1.2.4	listen	3
1.2.5	accept	4
1.2.6	send[tcp]	4
1.2.7	sendto[udp]	4
1.2.8	recv[tcp]	5
1.2.9	recvfrom[udp]	5
1.2.10	close	5
1.2.11	shutdown	6
1.3	辅助函数	6
1.3.1	inet_addr	6
1.3.2	inet_ntoa	6
1.3.3	bzero	7
1.3.4	htons	7
1.3.5	inet_aton	7

*Email:haojunyu2012@gmail.com

1 mysql的使用心得

1.1 mysql命令

1.1.1 用户管理

启动服务 `service mysql start(ubuntu)`

登录 `mysql -u username -ppassword`

退出 `exit`

创建用户 `insert into mysql.user(Host,User>Password) values("localhost","username","password(password))"`

给用户授予数据库的全部权限 `grant all privileges on dbname.* to username@localhost identified by 'password'`

给用户授予数据库的部分权限 `grant insert,delete,update,select on dbname.tablename to username@localhost identified by 'password'`

刷新系统权限表 `flush privileges`

删除用户 `delete * from mysql.user where User="username" and Host="localhost"`

修改用户密码 `update mysql.user set password=password("password") where User="username" and Host="localhost"`

1.1.2 数据库管理

备份数据库 `mysqldump -u username -ppassword dbname [tablename] > C : \file.sql`

导入数据库 `mysql -u username -ppassword dbname < C : \file.sql`

执行sql脚本 `source createdb.sql`

1.1.3 操作数据库

- 增

建立数据库 `create database dbname;`

建立表 `create table if not exists tablename(userid INT, userName VARCHAR(50));`

复制表 `create table tablename2 select * from tablename1 where 1 <= 1`

创建(唯一)索引 `create [unique] index idxId on tablename(colname)`

- 删

数据库 `drop database dbname`

数据库中的表 `drop table tablename`

清空表中记录 `delete from tablename`

删除索引 `drop index idxId on tablename`

- 改

修改(增加)多个字段 `alter tablename add column fieldname1 datatype1,add column fieldname2 datatype2`

修改字段类型 `alter table tablename modify colname coltype`

修改字段名称以及类型 `alter tabel tablename change colname newcolname datatype`

更行表中记录 `update tablename set userName='hgy' where userId='0005'`

- 查

显示数据库 `show databases`

使用数据库 `use dbname`

显示表格 `show tables`

显示表格列的属性 `show columns from tablename`

查询时间 `select now()`

查询当前用户 `select user()`

查询数据库版本 `select version()`

查询当前使用的数据库 `select database()`

1.1.4 内置函数

合并字段显示 `select concat(userId,':',userName,"=") from tablename`

选择10-20行 `select * from tablename order by colname limit 9,10`

1.2 C访问和操作mysql

1.2.1 准备(ubuntu)

- 安装mysql-server

```
1 hgy@gy :~$ sudo apt-get install mysql-server -5.5
```

- 安装用于连接到数据库和执行数据库查询的库文件mysqlclient.

```
1 hgy@gy :~$ sudo apt-get install libmysqlclient-dev
```

安装成功后,相关文件如下:

头文件(mysql.h) 位于/usr/include/mysql目录下;

库文件(libmysqlclient.so) 位于/usr/lib/mysql和/usr/lib目录下

1.2.2 操作数据库

- 控制类函数

`mysql_init` 初始化MYSQL对象

`mysql_options` 设置连接选项

`mysql_real_connect` 连接到mysql数据库

`mysql_real_escape_string` 将查询串合法化
`mysql_query` 发出一个以空字符结束的查询串
`mysql_real_query` 发出一个查询串
`mysql_store_result` 一次性传送结果
`mysql_use_result` 逐行传送结果
`mysql_free_result` 释放结果集
`mysql_change_user` 改变用户
`mysql_select_db` 改变默认数据库
`mysql_debug` 送出调试信息
`mysql_dump_debug_info` 转储调试信息
`mysql_ping` 测试数据库是否处于活动状态
`mysql_shutdown` 请求数据库SHUTDOWN
`mysql_close` 关闭数据库连接

- 信息获取类函数

`mysql_character_set_name` 获取默认字符集
`mysql_get_client_info` 获取客户端信息
`mysql_host_info` 获取主机信息
`mysql_get_proto_info` 获取协议信息
`mysql_get_server_info` 获取服务器信息
`mysql_info` 获取部分查询语句的附加信息
`mysql_stat` 获取数据库状态
`mysql_list_dbs` 获取数据库列表
`mysql_list_tables` 获取数据表列表
`mysql_list_fields` 获取字段列表

- 行类操作函数

`mysql_field_count` 获取字段数
`mysql_affected_rows` 获取受影响的行数
`mysql_insert_id` 获取AUTO_INCREMENT列的ID值
`mysql_num_fields` 获取结果集中的字段数
`mysql_field_tell` 获取当前字段位置
`mysql_field_seek` 定位字段
`mysql_fetch_field` 获取当前字段
`mysql_fetch_field_direct` 获取指定字段
`mysql_fetch_fields` 获取所有字段的数组
`mysql_num_rows` 获取行数
`mysql_fetch_length` 获取行长度
`mysql_row_tell` 获取当前行位置

`mysql_row_seek` 行定位
`mysql_data_seek` 行定位
`mysql_fetch_row` 获取当前行

- 线程类操作函数

`mysql_list_processes` 返回所有线程列表
`mysql_thread_id` 返回当前线程ID
`mysql_thread_safe` 是否支持线程方式
`mysql_kill` 杀死一个线程

- 出错处理类函数

`mysql_errno` 获取错误号
`mysql_error` 获取错误信息

- c程序样例

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <mysql.h>
4
5 int main()
6 {
7     MYSQL mysql;
8     MYSQL_RES *res;
9     MYSQL_ROW row;
10    char sql[50];
11    int t;
12
13    sprintf(sql, "select * from USER"); // 定义执行的语句SQL
14    mysql_init(&mysql); // 初始化结构mysql
15    if (!mysql_real_connect(&mysql, "localhost", "
        virtual_desktop", "virtual_desktop", "
        virtual_desktop", 3306, NULL, 0))
16        printf("\数据库连接发生错
            误n:%s\n", mysql_error(&mysql));
17    else
18        printf("\数据库连接成功n!\n");
19
20    t = mysql_real_query(&mysql, sql, (unsigned int)
        strlen(sql));
21                                // 在服务器上执行定义的语句SQL
22    if (t)
23        printf("查询语句错误: %s\n", mysql_error(&mysql));
24    else
25    {
26        res = mysql_store_result(&mysql);
27        while (row = mysql_fetch_row(res))
28        {
29            for (t = 0; t < mysql_num_fields(res); t++)
30                printf("%s", row[t]);
31            printf("\n");
32        }
33    }
```

```

32     }
33 }
34 printf("释放结果集的空间...\n");
35 mysql_free_result(res);
36
37 mysql_close(&mysql); // 释放数据库
38 printf("sql_error!_%s\n", mysql_error(&mysql));
39
40 return 0;
41 }

```

1.2.3 编译&连接

- 编译

```
1 hjy@jy : ~$ gcc -c mysql.c -I /usr/include/mysql
```

-I 指定头文件目录

- 连接

```
1 hjy@jy : ~$ gcc mysql.o -o main -L /usr/lib -lmysqlclient
```

-L 指定库文件路径

-lmysqlclient libmysqlclient.so库文件

库文件类型简介

.o 编译的目标文件

.a 静态库,其实就是把若干.o文件打了个包

.so 动态链接库(共享库)

.lo 使用libtool编译出来的目标文件,其实就是在.o文件中添加了一些信息

.la 使用libtool编译出来的库文件,其实就是个文本文件,记录同名动态库和静态库的相关信息