**1. 编写 RPC 接口定义文件**

创建一个 .x 文件，例如 date.x，用于定义 RPC 接口。以下是一个简单的示例，实现一个获取当前日期的服务：

*/\* date.x \*/*

struct Date {

int year;

int month;

int day;

};

program DATE\_PROG {

version DATE\_VER {

struct Date GET\_DATE(void) = 1; //过程号为 1，用于在版本 DATE\_VER 中唯一标识该过程

} = 1; //版本号设为1，以便在接口升级时保持向后兼容

} = 0x20000002; //程序号为 0x20000002，这是一个唯一标识符，用于区分不同的 RPC 程序

**2. 使用 rpcgen 生成代码**

运行以下命令生成 RPC 相关代码：

rpcgen -C -a date.x

这将生成以下文件：

* date.h：头文件，包含 RPC 接口定义。
* date\_clnt.c：客户端存根代码。
* date\_svc.c：服务端存根代码。
* date\_xdr.c：XDR（外部数据表示）代码，用于数据序列化。
* date\_client.c：客户端示例代码。
* date\_server.c：服务端示例代码。

**3. 修改客户端和服务端代码**

**修改服务端代码 date\_server.c**

在 date\_server.c 中，添加服务端逻辑，用于获取当前日期并返回：

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include "date.h"

struct Date \*get\_date\_1\_svc(void \*argp, struct svc\_req \*rqstp) {

static struct Date result;

time\_t now = time(NULL);

struct tm \*local\_time = localtime(&now);

result.year = local\_time->tm\_year + 1900; *// tm\_year 是从 1900 开始的*

result.month = local\_time->tm\_mon + 1; *// tm\_mon 是从 0 开始的*

result.day = local\_time->tm\_mday;

return &result;

}

**修改客户端代码 date\_client.c**

在 date\_client.c 中，添加客户端调用逻辑，用于请求服务端获取日期并打印结果：

#include <stdio.h>

#include <stdlib.h>

#include "date.h"

int main(int argc, char \*argv[]) {

if (argc != 2) {

fprintf(stderr, "Usage: %s <hostname>\n", argv[0]);

exit(1);

}

CLIENT \*clnt;

struct Date \*result;

clnt = clnt\_create(argv[1], DATE\_PROG, DATE\_VER, "udp");

if (clnt == NULL) {

clnt\_pcreateerror(argv[1]);

exit(1);

}

result = get\_date\_1(NULL, clnt);

if (result == NULL) {

clnt\_perror(clnt, "call failed");

} else {

printf("Current Date: %d-%02d-%02d\n", result->year, result->month, result->day);

}

clnt\_destroy(clnt);

return 0;

}

**4. 编译和运行**

**编译代码**

使用以下命令编译客户端和服务端代码：

gcc -o date\_server date\_server.c date\_svc.c date\_xdr.c

gcc -o date\_client date\_client.c date\_clnt.c date\_xdr.c

**运行服务端**

在终端中运行服务端程序：

./date\_server

**运行客户端**

在另一个终端中运行客户端程序，指定服务端的主机名或 IP 地址：

./date\_client 127.0.0.1

**5. 测试**

客户端会向服务端发送请求，服务端返回当前日期。客户端会打印出类似以下的结果：

Current Date: 2025-03-11