Practical Work 2: RPC File Transfer

Pham Huu Minh - 23BI14302

2025-12-05

Objective

Upgrade the existing TCP file transfer implementation to a Remote Procedure Call (RPC) based service that allows a client to request a file from the server and save it locally.

Design

RPC Interface

- Program: `FILE_TRANSFER_PROG` with version `FILE_TRANSFER_VERS`.
- Procedure: `GET_FILE(filename)` returns `file_result` containing `status` and `data`.
- Constraints: Maximum file size is 1 MB.

Data Flow

1. Client sends file name via RPC call.
2. Server reads file, enforces size limit, and returns bytes or an error status.
3. Client writes received bytes to local disk.

Figure: System Overview

Organization

- `file_transfer.x`: RPC interface definition (rpcgen input).
- `file_transfer_clnt.c`, `file_transfer_svc.c`, `file_transfer_xdr.c`: Generated stubs/XDR.
- `client.c`: RPC client executable.
- `server_impl.c`: Server-side implementation of `GET_FILE`.
- `test.txt`, `test_copy.txt`: Sample files for validation.

Figure: Module Layout

Implementation

Server Core (excerpt)

```
file_result *get_file_1_svc(filename_t *name, struct svc_req *rqstp)
{
    static file_result res;
    FILE *f;
    long len;
    char *buf;

    if (res.data.filedata_t_val) {
        free(res.data.filedata_t_val);
        res.data.filedata_t_val = NULL;
        res.data.filedata_t_len = 0;
    }
    res.status = 0;

    f = fopen(*name, "rb");
    if (!f) { res.status = errno; return &res; }

    fseek(f, 0, SEEK_END);
    len = ftell(f);
    if (len < 0 || len > MAXFILESIZE) { res.status = EFBIG; fclose(f); ret

    buf = malloc(len);
    rewind(f);
    if (fread(buf, 1, len, f) != (size_t)len) { res.status = EIO; free(buf
    fclose(f);

    res.data.filedata_t_val = buf;
    res.data.filedata_t_len = (u_int)len;
    return &res;
}
```

Client Core (excerpt)

```
int main(int argc, char *argv[])
{
    CLIENT *cl = clnt_create(argv[1], FILE_TRANSFER_PROG, FILE_TRANSFER_VE
    file_result *res = get_file_1(&argv[2], cl);
    FILE *fp = fopen(argv[3], "wb");

    fwrite(res->data.filedata_t_val, 1, res->data.filedata_t_len, fp);
    fclose(fp);
    clnt_destroy(cl);
    return 0;
}
```

Task Division

This is an individual work completed by Pham Huu Minh (23BI14302). All tasks were completed individually:

- RPC interface design (`file_transfer.x`) and server implementation.
- Client implementation and testing.
- Build scripts, documentation, and report.

Testing

- Start RPC server: `./rpc_server`
- Run client: `./rpc_client <host> test.txt downloaded.txt`
- Verify: compare `test.txt` and `downloaded.txt` (e.g., `diff` or checksum).