

1 Servidor DS - Organização de dados

O servidor DS instalado na máquina ‘tejo’ do laboratório LT5 usa uma estrutura de directorias para armazenar toda a informação referente ao protocolo de forma persistente entre sessões.

Na directoria de trabalho do servidor DS encontram-se duas directorias:

/USERS

/GROUPS

1.1 A directoria USERS

A directoria USERS contém informação sobre os utilizadores registados e os utilizadores com sessões activas. Sob a directoria ‘USERS’ existe uma directoria por cada utilizador registado.

/USERS/uid1 - directoria do user com identificação uid1

 /uid2 - directoria do user com identificação uid2

Cada directoria uid(n) (a designação uid(n) é composta sempre por 5 algarismos) é criada quando o utilizador emite o comando *reg* para se registar.

Dentro de cada directoria `uid(n)` existe um ficheiro designado

`uid(n)_pass.txt`

contendo a password do utilizador registado.

Se o utilizador tiver uma sessão activa (iniciada com o comando *login*) existe também dentro da directoria `uid(n)` um ficheiro designado

`uid(n)_login.txt`

O ficheiro **`uid(n)_login.txt`** é eliminado quando o utilizador efectua *logout*.

Quando o utilizador emite o comando *unregister*, a directoria `uid(n)` (em `/USERS`) é eliminada juntamente com os ficheiros que possa conter nesse momento.

1.2 A directoria GROUPS

A directoria GROUPS contém informação sobre os grupos criados, os seus nomes, os utilizadores que lhes estão associados (com *subscribe*) e as mensagens publicadas e acessíveis aos utilizadores associados ao cada grupo.

Assim, a directoria GROUPS contém uma directoria por cada grupo criado:

`/GROUPS/gid1` - directoria do grupo com identificação `gid1`

`/gid2` - directoria do grupo com identificação `gid2`

Cada directoria `gid(n)` contém um ficheiro designado

`gid(n)_name.txt`

contendo o nome do grupo em causa. E ainda um ficheiro com designação **`uid(n).txt`** por cada utilizador que estiver associado ao grupo (com *subscribe*).

A directoria genérica gid(n) tem uma sub-directoria designada MSG contendo as mensagens publicadas para o grupo gid(n).

A estrutura de directorias para armazenamento das mensagens publicadas é exemplificada como segue:

/GROUPS/gid1/MSG/mid1 - directoria da mensagem do grupo gid1 com identificação mid1.
/mid2 - directoria da mensagem do grupo gid1 com identificação mid2.

Cada directoria de mensagem mid(n) é criada quando um utilizador faz *post*.

Cada directoria mid(n) contém obrigatoriamente dois ficheiros:

- Um ficheiro designado **T E X T.txt** contendo o texto da mensagem.
- Um ficheiro designado **A U T H O R.txt** contendo o UID do autor da mensagem.

Cada directoria mid(n) pode conter ainda, opcionalmente, um ficheiro adicional que possa ter sido publicado com o *post* da mensagem respectiva.

Exemplifica-se a seguir um diagrama em árvore sob a directoria GROUPS para o grupo com identificação '01':

/GROUPS/01/01_name.txt
/GROUPS/01/11111.txt
/GROUPS/01/22222.txt
/GROUPS/01/MSG/0001/A U T H O R.txt
/GROUPS/01/MSG/0001/T E X T.txt
/GROUPS/01/MSG/0002/A U T H O R.txt
/GROUPS/01/MSG/0002/T E X T.txt
/GROUPS/01/MSG/0002/photo.jpg

- O ficheiro **01_name.txt** contém o nome do grupo '01'.
- O ficheiro 11111.txt indica que o utilizador com identificação 11111 está associado ao grupo '01'.
- O ficheiro 22222.txt indica que o utilizador com identificação 22222 está associado ao grupo '01'.
- A directoria /MSG/0001 foi criada para armazenar a primeira mensagem do grupo '01'. Esta directoria contém o ficheiro com o texto da mensagem e o ficheiro com a identificação do seu autor.
- A directoria /MSG/0002 foi criada para armazenar a segunda mensagem do grupo '01'. Esta directoria contém o ficheiro com o texto da mensagem, o ficheiro com a identificação do seu autor e ainda um ficheiro designado **photo.jpg** carregado com a mensagem em causa aquando da emissão do comando *post* na aplicação user.

Os ficheiros **A U T H O R.txt** e **T E X T.txt** contêm propositadamente espaços nas suas designações para evitar que um ficheiro opcional carregado com o comando *post* se lhes possa sobrepor por coincidência de designação.

2 USERS e GROUPS especiais para testes

Com a finalidade de controlar e preservar exclusivamente para testes de consulta alguma da informação existente na base de dados do servidor DS, aos UID e aos GID começados por zero estão vedadas todas as operações que possam alterar os dados de utilizadores e de grupos, nomeadamente a inserção de novas mensagens por esses UID e para esses GID.

2.1 USERS especiais

Para os UID começados por zero pré-existent na base de dados do servidor DS na máquina 'tejo', apenas são possíveis operações que não impliquem alterações na base de dados.

- *login* - é efectuada a verificação da password, mas não é mantido registo sobre a sessão.
- *logout* - é possível mas não produz efeito.

- *groups*
- *my_groups*
- *select* - comando de efeito apenas local
- *ulist*
- *retrieve*

2.2 GROUPS especiais

Existem nove grupos especiais na base de dados do servidor DS numerados de 01 a 09 para os quais só são possíveis as seguintes operações:

- *groups*
- *my_groups*
- *select*
- *ulist*
- *retrieve*

3 Complementos de programação para tratamento de directorias e ficheiros

Com a finalidade de auxiliar a concretização de uma estrutura de dados como aquela que se apresentou na secção 1, ilustram-se aqui algumas funções usadas pelo servidor DS para a criação, leitura e remoção de directorias e eliminação de ficheiros usando funções de interacção com o *filesystem* definidas na linguagem C.

As funções aqui exemplificadas podem ser generalizadas para utilizações semelhantes no contexto do projecto.

3.1 Criação de directoria de utilizador após recepção de mensagem REG

```
#include <unistd.h>

int CreateUserDir(char *UID)
{
    char user_dirname[20];
    int ret;

    sprintf(user_dirname,"USERS/%s",UID);
    ret=mkdir(user_dirname,0700);
    if(ret== -1)
        return(0);
    return(1);
}
```

3.2 Eliminação de directoria de utilizador após recepção de mensagem UNR

```
#include <unistd.h>

int DelUserDir(char *UID)
{

```

```

char user_dirname[20];
sprintf(user_dirname,"USERS/%s",UID);

if(rmdir(user_dirname)==0)
    return(1);
else
    return(0);
}

```

3.3 Leitura do conteúdo de uma directoria

A função ilustrada abaixo lê o conteúdo da directoria ‘GROUPS’ para preencher uma lista de todos os grupos existentes na base de dados. Para cada grupo, lê também o ficheiro existente na subdirectoria respectiva contendo o nome do grupo. No final da execução, a estrutura cujo ponteiro se designa por ‘list’, fica preenchida com todos os GID de todos os grupos e os respectivos nomes.

```

#include <dirent.h>

int ListGroupsDir(GROUPLIST *list)
{
    DIR *d;
    struct dirent *dir;
    int i=0;
    FILE *fp;
    char GIDname[30];

```

```

list->no_groups=0;

d = opendir("GROUPS");

if (d)
{
    while ((dir = readdir(d)) != NULL)
    {
        if(dir->d_name[0]=='.')
            continue;
        if(strlen(dir->d_name)>2)
            continue;
        strcpy(list->group_no[i], dir->d_name);
        sprintf(GIDname,"GROUPS/%s/%s_name.txt",dir->d_name,dir->d_name);
        fp=fopen(GIDname,"r");
        if(fp)
        {
            fscanf(fp,"%24s",list->group_name[i]);
            fclose(fp);
        }
        ++i;
        if(i==99)
            break;
    }
}

```



```

    }
    list->no_groups=i;
    closedir(d);
}
else
    return(-1);

if(list->no_groups>1)
    SortGList(list);

return(list->no_groups);
}

```

3.4 Eliminação de ficheiro de password após recepção de mensagem OUT

A função DelPassFile é invocada pelo servidor DS no contexto de uma operação *logout* para eliminar o ficheiro que indica o estado da sessão do utilizador.

```

#include <unistd.h>

int DelPassFile(char *UID)
{
    char pathname[50];
    sprintf(pathname,"USERS/%s/%s_pass.txt",UID,UID);

    if(unlink(pathname)==0)

```

```
        return(1);  
    else  
        return(0);  
}
```