Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Unidade Curricular de Segurança de Sistemas Informáticos

Ano Letivo de 2023/2024

Serviço Local de Troca de Mensagens (TP2)

Duarte Leitão(a100550) Diogo Barros(a100600) Bento Guimarães(a96296)

12 de maio de 2024

SSI

Data de Receção	
Responsável	
Avalição	
Observações	

Serviço Local de Troca de Mensagens (TP2)

Duarte Leitão(a100550) Diogo Barros(a100600) Bento Guimarães(a96296)

12 de maio de 2024

Índice

1 Introdução

Pretende-se desenvolver um serviço de conversação entre utilizadores locais de um sistema Linux. A este respeito, deve-se considerar que os utilizadores já existem no sistema operativo (o serviço não terá necessidade de os criar).

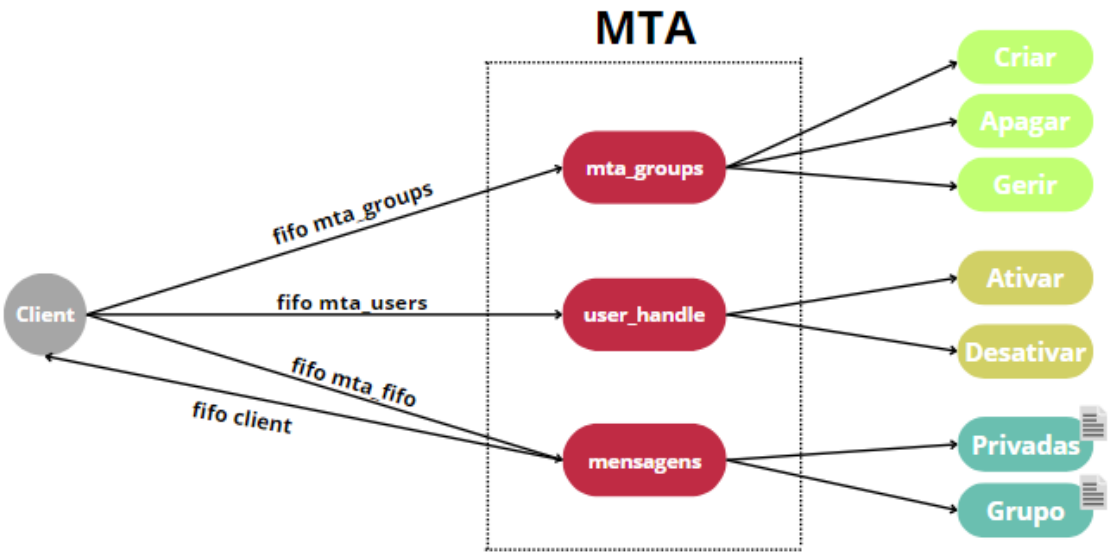
O serviço deverá suportar o envio de mensagens para um utilizador, bem como a sua leitura pelo respetivo destinatário. A leitura de mensagens pode ser feita de forma síncrona ou assíncrona.

Deverá também suportar a gestão de utilizadores do serviço, ou seja, um utilizador pode "entrar"e "sair"do serviço.

O serviço deverá ainda suportar a noção de grupos privados de conversação. Nesse sentido, deverá oferecer mecanismos para criação e remoção de grupos, e de gestão dos seus membros. É necessário ter em conta que o criador do grupo é responsável por toda esta administração.

Deve ser tida um conta a segurança do serviço. As noções de grupos de utilizadores do sistema e permissões associadas a objetos, presentes no Linux, devem estar presentes na implementação do serviço.

2 Arquitetura da solução



2.1 MTA

O MTA é o programa responsável pela gestão de utilizadores, gestão de grupos e entregas de mensagens. De forma a suportar cada um destes domínios do serviço, o programa principal cria um processo filho e um fifo, ou seja, na iniciação do serviço são criados três processos filho e três fifos.

2.1.1 Gestão de utilizadores

No domínio da gestão de utilizadores, o objetivo é permitir que um utilizador do sistema seja ativado ou desativado. Por outras palavras, este processo reúne as condições necessárias para um cliente interagir com o serviço e toma as medidas necessárias para marcar a "saída"de um utilizador do serviço.

As funcionalidades deste domínio usam a estrutura de dados **User_request**, que contém o username do utilizador que realiza o pedido e uma flag, que indica o tipo de pedido. Um

pedido de ativação de utilizador é marcado com flag = 0 e um pedido de desativação de utilizador é marcado com flag = 1. Todos os pedidos são recebidos através do fifo **client_fifo** e processados com base no valor da flag. Por cada pedido recebido é criado um processo filho.

Caso o pedido seja de ativação de utilizador, será criado um ficheiro csv que servirá de caixa de correio (*username*_file.csv), um fifo para o utilizador que irá suportar a comunicação síncrona (*username*_fifo) e por fim o utilizador é adicionado ao grupo mta_users, sendo este o grupo que contém todos os utilizadores do serviço (não é um grupo de conversação).

A desativação do utilizador consiste no processo inverso à ativação: eliminar a caixa de correio, eliminar o fifo e remover o utlizador do grupo mta_users.

2.1.2 Gestão de grupos de conversação

A funcionalidade de gestão de grupos de conversação do MTA, permite que os utilizadores criem ou eliminem grupos, e que estes possam fazer a gestão dos seus utilizadores (adicionar ou remover).

Estas funcionalidades usam a estrutura de dados **Group_request**, que contém o username de quem faz o pedido, o nome do grupo que se pretende gerir, um array com nomes de utilizador (a adicionar ou a remover), o número de utilizadores do array e uma flag que indica o tipo de pedido. A flag = 0 indica que é um pedido de criação de grupo. Flag = 1 indica que é um pedido para eliminar o grupo. Flag = 2 indica que se pretende adicionar utilizadores a um grupo. Flag = 3 indica que se pretende remover utilizadores de um grupo. Por cada pedido recebido é criado um processo filho.

Na criação de grupos, caso o grupo não existir, é criado o grupo e são adicionados todos os membros indicados pelo utilizador que fez o pedido de criação. É ainda criado um ficheiro csv (<nome do grupo>_group_file.csv) para guardar as mensagens de grupo, à semelhança do que acontece para os utilizadores individuais.

Já para eliminar um grupo, primeiro verifica-se se o grupo não se chama mta_users (grupo de todos os utilizadores do sistema). De seguida, verifica-se se o utilizador que fez o pedido de eliminação é o dono do grupo. Caso estas condições se verificarem, o grupo e o seu csv são eliminados.

O processo de adicionar e remover membros é bastante semelhante. Caso o utilizador que fez o pedido seja o dono do grupo, o processo de adição ou remoção dos membros especificados no pedido irá proceder.

Todos estes pedidos de gestão de grupos são feitos através do fifo **mta_groups**.

2.1.3 Gestão de mensagens

A funcionalidade principal do MTA é a receção, armazenamento e entrega de mensagens privadas ou de grupo. Esta funcionalidade é suportada pela estrutura de dados **Message**. A mensagem contém o username do remetente, o username do destinatário, o nome do grupo, um assunto e o contúdo da mensagem. Visto que esta estrutura suporta tanto as mensagens privadas como as mensagens de grupo, o MTA deve analisar os valores *receiver* e *group* antes de processar a mensagem. Caso exista um *receiver*, mensagem é processa como mensagem privada. Caso exista um *group*, a mensagem é processada com mensagem de grupo.

No caso das mensagens privadas, será gerado um número aleatório para servir de id the mensagem e um time-stamp. De seguida o MTA irá escrever a mensagem no csv do *receiver*. Por fim, a mensagem recebida será enviada para o utilizador pelo fifo *receiver*_fifo (fifo criado para o utilizador na ativação). Por cada mensagem privada recebida é criado um processo filho.

Já nas mensagens de grupo, também é gerado o id e o time-stamp. A mensagem será no entanto escrita no ficheiro do grupo. De seguida, o MTA irá usar o nome do grupo para obter os membros do grupo. Desta forma pode enviar para cada membro a mensagem através do fifo de cada utilizador. Por cada mensagem de grupo recebida é criado um processo filho. Este processo criado irá, por sua vez, criar um processo filho por cada utilizador presente no grupo para enviar a mensagem através do fifo do utilizador.

2.2 Client

O clinet é o programa que permite qualquer utilizador do sistema interagir com o serviço de mensagens. Este programa tem dois processos associados. O primeiro fica à espera de comandos que permitem realizar as operações disponíveis no serviço. O segundo espera por mensagens que chegam através do fifo do utilizador. Antes do client poder executar qualquer tipo de operação, este vai buscar o seu username, posteriormente será usado nas operações disponíveis no serviço. Estas operações são invocadas pelos comandos. Os comandos não necessitam de argumentos. Estes são depois obtidos nas funções relativas às operações através de input do utilizador. Em relação ao processo que espera por mensagens vindas do fifo do utilizador, por cada mensagem recebida é criado um processo filho que irá mostrar ao utilizador a mensagem.

No client estão disponíveis os seguintes comandos:

- **activate**: adiciona o utilizador no serviço
- **deactivate**: remove o utilizador do serviço
- **send**: envia uma mensagem privada

- **send_grp**: envia uma mensagem de grupo
- **inbox**: caixa de mensagens do utilizador
- **get_msg**: vai buscar a mensagem com um certo ID
- **group_inbox**: caixa de mensagens de um grupo que o utilizador pertence
- **get_grp_msg**: vai buscar a mensagem com um certo ID enviada para um dado grupo
- **groups**: grupos a que o utilizador pertence
- **group_members**: mostrar membros de um grupo que o utilizador pertence
- **create_group**: cria um grupo de conversação
- **delete_group**: apaga um grupo
- **add_users_to_group**: adiciona utilizadores a um grupo
- **remove_users_from_group**: remove utilizadores de um grupo

3 Segurança do serviço

Sobre questões de segurança do serviço é relevante mencionar que o MTA corre com sudo, visto que usa várias vezes a função system para correr comandos que utilizam sudo. O mesmo não se verifica para o programa client. O que o client pode ou não fazer/aceder está controlado por permissões.

3.1 Permissões dos fifos

As permissões associadas aos fifos têm um grande peso na segurança e bom funcionamento do serviço. Estas permissões são geridas com acl.

O fifo **client_fifo** tem permissões de escrita e leitura para qualquer utilizador do sistema, visto que a sua finalidade é permitir enviar pedidos de ativação e desativação ao MTA. Se um utilizador quiser ingressar no serviço, deve poder fazê-lo sem grandes restrições.

Já para os fifos **mta_fifo** e **mta_groups**, existe algumas restrições. Para utilizar estes fifos é necessário ser o seu owner ou pertencer ao grupo **mta_users**. Os membros deste grupo têm permissões de escrita e leitura. Estas restrições foram implementadas, porque faz sentido que um utilizador que queira mandar mensagens ou usar grupos seja reconhecido como um membro do serviço. Esse é o objetivo do grupo mta_users.

No caso dos fifos dos utilizadores (fifos utilizados na comunicação síncrona com nome *username*_fifo), foram atribuidas permissões de leitura e escrita ao seu owner e ao utilizador a que se referem. Quando um utlizador é criado no sistema, é também criado um grupo com o username do utilizador. São atribuídas as devidas permissões do fifo a esse grupo, logo, apenas o MTA e o utlizador conseguem escrever nesse fifo.

3.2 Acesso a caixas de correio

Visto que existem vários ficheiros que servem de caixa de correio para utilizadores ou grupos, faz sentido restringir o acesso a estes ficheiros para alcançar algum nível de privacidade entre os utilizadores. São então atribuídas permissões com acl.

No caso do ficheiro csv de cada utilizador, o processo é semelhante ao do fifo do utilizador.

Depois da criação do ficheiro, são atribuídas permissões de leitura e escrita ao owner e ao grupo com nome igual ao username do utilizador. Desta forma apenas o utilizador e o MTA conseguem aceder a este ficheiro.

Por exemplo, podemos testar com o comando **cat** que o utilizador manel não consegue ler o ficheiro do utilizador ze:

```
manel@vivobook:/home/user/2324-G15/TPs/TP2$ cat ze_file.csv
cat: ze_file.csv: Permissão recusada
manel@vivobook:/home/user/2324-G15/TPs/TP2$
```

No caso do csv de um grupo de conversação, são atribuídas permissões de leitura e escrita ao owner e ao grupo que vai usar o ficheiro como caixa de correio. Desta forma apenas o MTA e os membros do grupo conseguem aceder a este ficheiro.

Podemos realizar o teste anterior para os ficheiros de grupo. Pelas figuras que se seguem podemos observar os grupos do utilizador manel e do utilizador ze com o comando **groups** do client:

```
manel@vivobook:/home/user/2324-G15/TPs/TP2$ ./client
mta_client> groups
----------------------------------------
Grupos:
----------------------------------------
mta_users (1006)
grupo1 (1011)
----------------------------------------
mta_client>
```

```
ze@vivobook:/home/user/2324-G15/TPs/TP2$ ./client
mta_client> groups
----------------------------------------
Grupos:
----------------------------------------
mta_users (1006)
----------------------------------------
mta_client>
```

Pela figura a seguir conseguimos verificar que o utilizador ze não consegue ler o ficheiro de um grupo a que não pertence:

```
ze@vivobook:/home/user/2324-G15/TPs/TP2$ cat grupo1_group_file.csv
cat: grupo1_group_file.csv: Permissão recusada
ze@vivobook:/home/user/2324-G15/TPs/TP2$
```

Vale a pena mencionar que, um utilizador que foi removido de um grupo, não consegue ler as mensagens enviadas para o mesmo no passado. Como já não pertence ao grupo, perde as permissões sobre o ficheiro do mesmo.

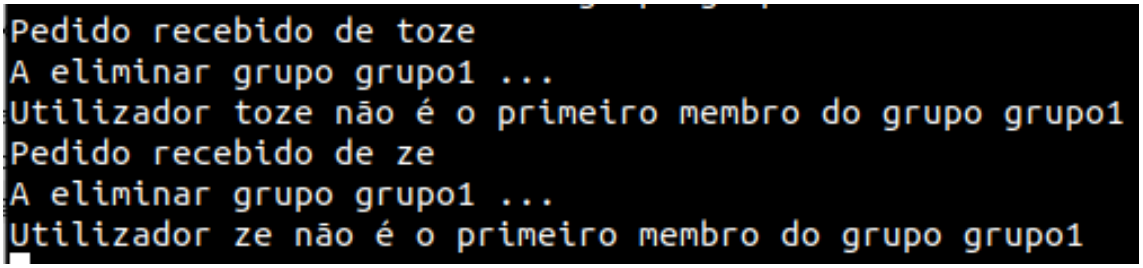
3.3 Gestão de grupos

Qualquer utilizador presente no grupo mta_users, o grupo de utilizadores do serviço, consegue criar um grupo. Para os processos de eliminar o grupo e adicionar ou remover utilizadores já existem algumas restrições. Para além de o utilizador ter que estar presente no mta_users, será ainda verificado se este é o owner do grupo.

Para verificar quem é o owner do grupo, temos que obter a lista dos seus membros. O primeiro membro será o owner, visto que foi a primeira pessoa a ser adicionada ao grupo no seu processo de criação.

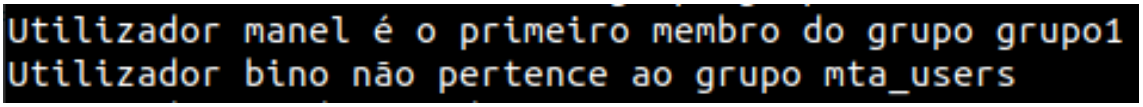
Vale a pena mencionar que não é possível criar um grupo com nome mta_users visto que este grupo já existe. É ainda impossível eliminar e adicionar ou remover utilizadores deste grupo.

Vamos verificar que um utilizador deve ser o owner de um grupo para o gerir. Para tal vamos usar o comando **delete_group** do client. Neste teste os utilizadores manel e toze pertence ao "grupo1", mas no entanto, o utilizador ze não pertence.

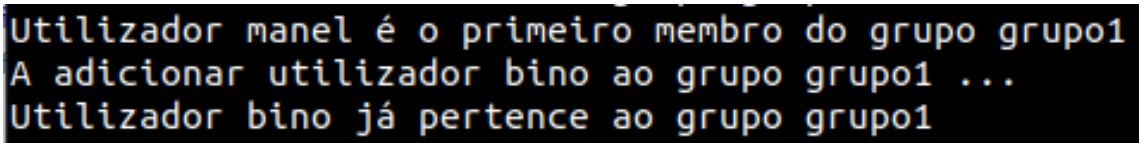


Pelos logs do MTA na figura anterior, podemos verificar que o toze e o ze não conseguiram apagar o grupo. Podemos então concluir que o MTA está a verificar corretamente que é o owner do grupo.

No processo de adição de membros a um grupo também existem restrições. Caso o utilizador não pertença ao grupo mta_users, não poderá ser adicionado a um grupo:



Caso o utilizador já pertença ao grupo, também não voltará a ser adicionado:



3.4 Envio de mensagens

As restrições associadas ao envio de mensagens estão associadas às restrições aplicadas aos fifos. Apenas utilizadores no grupo mta_users podem enviar mensagens privadas ou de grupo, visto que este fifo não tem permissões de escrita e leitura para outros utilizadores do sistema.

Já no caso das mensagens de grupo, é necessário que o sender pertença ao grupo para que a mensagem chegue à caixa de correio do grupo e aos destinatários.

4 Reflexão sobre a solução implementada

Na arquitetura deste serviço, no que toca ao domínio da segurança em particular, a criação do grupo mta_users mostrou ser bastante útil por todos os domínios do serviço. Para além de permitir controlar o acesso aos fifos, permitiu ainda definir com pode criar grupos ou enviar mensagens. No fundo foi uma forma de definir que utilizadores podem interagir com o serviço.

No que toca a caixas de correio individuais, considerou-se que a melhor forma de controlar o acesso às mesmas seria dar permissões de leitura apenas ao grupo com nome igual ao utilizador a que a caixa pertence, visto que quando um utilizador é criado, também é criado um grupo com o mesmo nome. Nas caixas de correio dos grupos o raciocínio foi semelhante, no entanto, agora é utilizado um grupo com todos os utilizadores.

No domínio da troca de mensagens, em particular nas mensagens privadas, consideramos relevante que cada utilizador tivesse o seu próprio fifo. Neste fifo podem ler e escrever os utilizadores que se encontram no grupo com nome igual ao utilizador. Assim apenas o utilizador e o admin do sistema podem aceder a este fifo. Desta forma o serviço consegue oferecer algum nível de segurança na entrega de forma síncrona das mensagens ao seu destinatário. Esta funcionalidade também é útil para a comunicação em grupo, visto que a mensagem de grupo, após recebida pelo MTA, vai ser enviada para cada utilizador do grupo de forma síncrona.

No que toca a armazenamento de mensagens em ficheiros, consideramos que deveria ser feita uma separação entre mensagens de grupo e mensagens privadas, ou seja, as mensagens dirigidas a um determinado utilizador não devem estar todas concentradas no mesmo lugar. Esta separação tornou mais fácil controlar o acesso a mensagens.

Em termos de arquitetura, do MTA em específico, decidimos dividir as funcionalidades por três processos diferentes, visto que cada um destes domínios está a lidar com estruturas de dados diferentes. Por exemplo as mensagens são processadas de forma diferente que os pedidos de grupo, logo, justifica-se a criação de diferentes processos que se especializem em processar cada uma destas estruturas de dados corretamente. Desta forma, estamos também a tornar o MTA modular.

5 Conclusão

O desenvolvimento deste serviço permitiu que o grupo aplicasse num ambiente prático todo o seu conhecimento sobre utilizadores, grupos de utilizadores e controlo do acesso a objetos, conceitos estes presentes no sistema Linux.

A existência de vários utilizadores no sistema não só necessitou fazer um controlo de acesso aos ficheiros usados pelo serviço, mas também criou a necessidade de incluir todos os utilizadores num grupo de utilizadores do serviço para garantir que os utilizadores que usam o serviço foram reconhecidos pelo mesmo.

Os grupos do Linux criaram ainda a possibilidade de introduzir grupos de conversação no serviço. Esta funcionalidade implicou que fossem também implementadas algumas medidas de segurança, nomeadamente atribuir as permissões devidas às caixas de correio dos grupos para manter alguma privacidade, e fazer um controlo da gestão de grupos, garantindo desta forma que esta é feita pelos utilizadores corretos.

Em suma, o desenvolvimento deste serviço implicou o desenhar uma arquitetura que, para além de obedecer a um conjunto de requisitos funcionais, deve ter implementados vários mecanismos de segurança pelos seus diferentes domínios.