Universidade do Minho

Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Segurança de Sistemas Informáticos

Ano Letivo de 2023/2024

Concordia - Serviço local de troca de mensagens

Grupo 42
A100758 - Hugo Arantes Dias
A100838 - Jorge Emanuel Matos Teixeira
A100749 - José Luís Fraga Costa

Índice

1 Introdução

O projeto visa desenvolver um serviço de conversação para utilizadores locais de um sistema Linux (Concordia), semelhante ao serviço já existente **qmail**. Esta abordagem é conhecida por ser bastante simples e modular, pelo que o objetivo deste projeto será seguir os mesmos princípios. O serviço permite o envio, receção e gestão de mensagens, bem como a criação e gestão de grupos de utilizadores. O foco do projeto inclui a implementação segura utilizando mecanismos de segurança do próprio sistema operativo.

2 Arquitetura Funcional

Como mencionado anteriormente, a arquitetura desenvolvida tem o **qmail** como base. Assim, descreveremos de seguida a forma como percepcionamos a nossa solução no projeto.

2.1 Programas e Processos

Primeiramente, abordaremos detalhadamente os programas desenvolvidos na nossa resolução:

2.1.1 concordia-ativar

Este programa cria uma "mailbox", isto é, uma diretoria chamada "<userName>Mail" na pasta home do utilizador atualmente logado no sistema. Neste programa o nome do utilizador é obtido e utilizado para criar o path completo necessário à adição da nova diretoria. Esta é criada com as permissões específicas para o utilizador logado.

2.1.2 concordia-desativar

Este programa remove a "mailbox" do utilizador. Resolve o caminho para a diretoria através do nome do utilizador e elimina-a removendo assim o utilizador do sistema.

2.1.3 concordia-enviar

Através deste programa o utilizador é capaz de enviar uma mensagem para um destinatário, seja ele um outro utilizador individual ou grupo a que pertença. Posteriormente, este processo será ainda mais detalhado.

2.1.4 concordia-ler

O programa **concordia-ler** permite ao utilizador ler uma mensagem de um ficheiro específico na sua diretoria. O programa aceita o ID da mensagem como argumento, abre o ficheiro correspondente, lê o conteúdo e imprime a mensagem no ecrã. No caso de ler de um grupo é feito da mesma forma, mas seguido do nome do grupo.

2.1.5 concordia-listar

Através deste programa o utilizador consegue listar mensagens da sua "mailbox". O programa por si imprime todas as mensagens não lidas da sua diretoria. Se este programa for executado com a flag **[-a]** são devolvidas todas as mensagens, lidas e por ler, da sua "mailbox". No caso de ser usada a flag **[-g]** o programa retorna todas as mensagens dos grupos em que o utilizador atual se encontra.

2.1.6 concordia-responder

Este programa possibilita ao utilizador atual responder a uma mensagem específica na sua diretoria. O programa aceita o ID da mensagem e o conteúdo da resposta como argumentos, lê o remetente original e envia da mesma forma que o “concordia-enviar”.

2.1.7 concordia-remover

O programa remove uma mensagem específica da diretoria "<nomeUtilizador>Mail" do utilizador atual. Ao aceitar o ID da mensagem como argumento, constrói o caminho para o ficheiro correspondente e remove o ficheiro. Se a remoção for bem-sucedida, imprime uma mensagem de confirmação; caso contrário, exibe uma mensagem de erro.

2.1.8 concordia-grupo-criar

Este programa cria um novo grupo de utilizadores no sistema. Utiliza comandos do sistema para criar o grupo e definir o utilizador atual como proprietário. Se o grupo for criado com sucesso, imprime uma mensagem de confirmação; caso contrário, exibe uma mensagem de erro.

2.1.9 concordia-grupo-remover

Este programa remove um grupo de utilizadores do sistema. O mesmo verifica se o utilizador atual é o proprietário do grupo a ser removido. Se bem-sucedida, o programa remove o grupo e imprime uma mensagem de sucesso; caso contrário, exibe uma mensagem de erro.

2.1.10 concordia-grupo-listar

Lista todos os membros de um grupo específico. O programa recebe o nome do grupo como argumento, verifica se o grupo existe e, em caso afirmativo, lista os membros do grupo, imprimindo cada um deles.

2.1.11 concordia-grupo-destinatario-adicionar

O programa **concordia-grupo-destinatario-adicionar** adiciona um utilizador a um grupo existente no sistema. O programa verifica se o utilizador a ser adicionado existe no sistema e se o utilizador atual é o proprietário do grupo. Se ambas as verificações forem bem-sucedidas, o programa adiciona o utilizador ao grupo e imprime uma mensagem de sucesso; caso contrário, exibe uma mensagem especificando qual o erro.

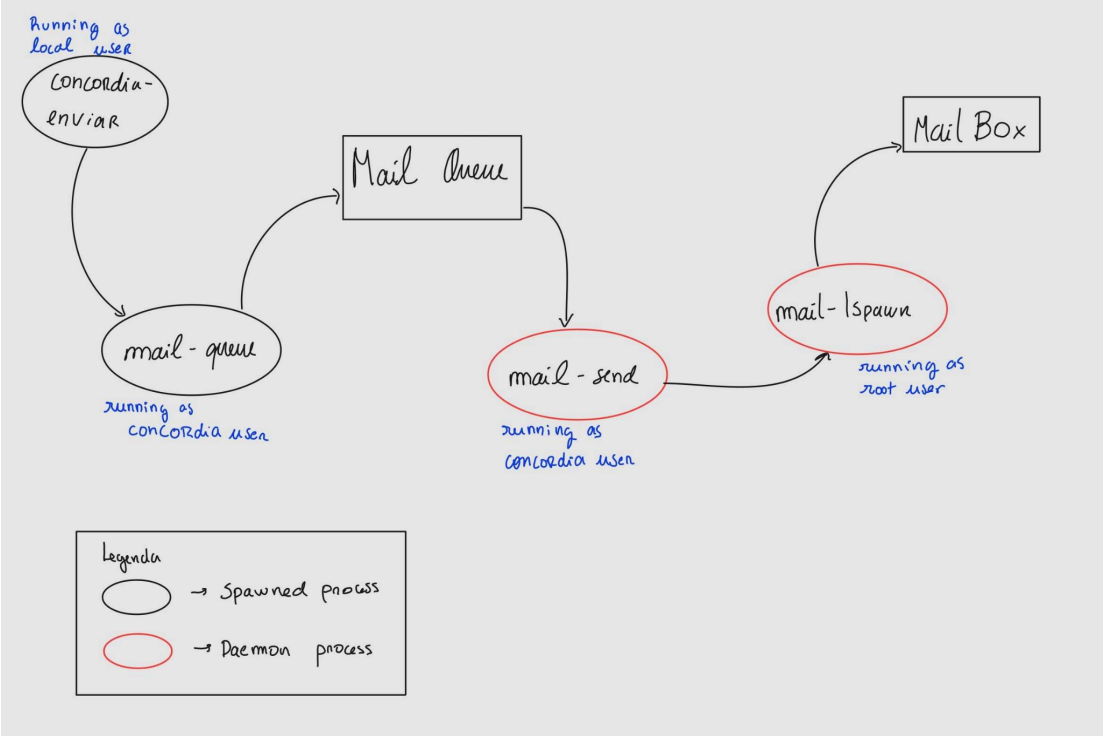
2.1.12 concordia-grupo-destinatario-remover

Por fim, este ficheiro remove um utilizador de um grupo existente. O programa verifica se o utilizador a ser removido existe no sistema e se o utilizador atual é o proprietário do grupo. Se as verificações forem

bem-sucedidas, o programa remove o utilizador do grupo e imprime uma mensagem de sucesso; caso contrário, exibe uma mensagem de erro.

2.2 Arquitetura de Envio

Na seguinte imagem podemos ver um esquema que representa a arquitetura do nosso projeto, associada ao funcionamento de envio de uma mensagem:



É possível descrever e separar em etapas da seguinte forma:

1. **concorda-enviar** - Este é o programa utilizado pelo utilizador do sistema destinado ao envio de mensagens. Na execução deste programa, é passado pelo próprio utilizador o nome do utilizador ou do grupo de destino e a mensagem a ser enviada como argumento. Após isto, é criado um processo-filho para a execução do processo **mail-queue** que recebe o destinatário e a mensagem a ser enviada através de um pipe.
2. **mail-queue** - Este programa executa sobre o utilizador "concordia" que é o único que possui permissões de escrita na diretoria "queue", está no caminho "/home/concordia/queue" e funciona como uma lista de espera, e é responsável por criar e adicionar à mesma um ficheiro de texto tendo em conta as informações recebidas. Este ficheiro tem o formato **<remetente>;<destinatário>;<mensagem>**, além disso tem como nome do ficheiro índice da mensagem.
3. **mail-send** - Ficheiro que executa como um Daemon (usando o systemd), sobre o utilizador "concordia" e tem a responsabilidade de continuamente verificar o conteúdo da diretoria

“queue”. Tendo permissões de leitura dos ficheiros contidos nesta diretoria, assim que um novo ficheiro é adicionado na fila de espera, o programa lê o conteúdo do mesmo e escreve o num “fifo”, que será lido pelo programa **mail-lspawn**. Logo de seguida, o programa remove o ficheiro da queue.

4. **mail-lspawn** - Por fim, este programa, que também funciona em modo Daemon, tem como função ler o fifo e guardar o conteúdo para a “mailbox” do destinatário correto. Como este processo executa sobre o utilizador “root”, consegue aceder à “mailbox” deste e escrever a mensagem. Esta diretoria tem genericamente o caminho “/home/<nomeUtilizador>/<nomeUtilizador>Mail”. Caso seja o grupo guarda na diretoria "/var/concordia/groups/<nomeGrupo>".

2.3 Estruturas de Dados

- **Ficheiros de Mensagens**: As mensagens são guardadas numa pasta, na diretoria home do utilizador em questão, com o nome “<username>Mail”.
- **Ficheiro de Mensagens por grupos**: Os grupos são guardados na diretoria "/var/concordia/groups", sendo criada dentro da anterior uma diretoria com o nome do grupo, onde vão ser guardadas as mensagens.
- **Fila de espera de Mensagens**: Todas as mensagens injetadas no sistema são guardadas numa diretoria com o nome “queue”. Sendo que as mensagens serão tratadas pelos processos indicados.

3 Decisões Tomadas

3.1 Controlo de Acesso

- **Permissões de Ficheiros**: Durante a criação tanto da diretoria como da cada ficheiro, são dadas permissões de acesso apenas ao utilizador em questão. Desta forma apenas os utilizadores a quem as mensagens pertencem têm o respetivo acesso. Os grupos funcionam de forma similar, isto é, apenas o grupo associado tem acesso à sua diretoria. Grupo este que foi previamente atribuído, utilizando funções do sistema Linux. Ainda sobre permissões sob ficheiros, é de realçar que apenas o utilizador denominado de "concordia" tem acesso à diretoria "queue" (fila de espera de mensagens).
- **Isolamento de Processos**: De forma a garantir maior segurança optamos por dividir o programa em vários processos, tal como explicado anteriormente, sendo que cada um apenas possui as permissões necessárias para a sua execução. Os processos que lidam com a "queue" executam sobre o utilizador "concordia". Na imagem abaixo é possível ver um exemplo das permissões dadas aos processos durante a compilação. Em que o "owner" do executável é mudado para o desejado com as permissões adequadas.

```
$(BINDIR)/$(QUEUE_NAME): $(SRCDIR)/$(QUEUE_NAME).c | $(BINDIR)
    $(CC) $(CFLAGS) $< -o $@
    sudo chown concordia:concordia $(BINDIR)/$(QUEUE_NAME)
    sudo chmod u+s $(BINDIR)/$(QUEUE_NAME)
```

3.2 Decisões sobre os Grupos

Quantos aos grupos optamos por usar a estrutura já existente no sistema linux, visto que facilitaria associar as permissões aos utilizadores para a respectiva pasta, quando estes são adicionados aos grupos.

- **Criação e remoção de grupos:** Quanto à remoção e criação mantivemos a restrição imposta pelo sistema linux, em que apenas utilizadores com permissões sudoers podem executar a criação destas ações.
- **Adicionar e Remover Utilizadores**: Apenas o dono do grupo consegue remover e adicionar utilizadores, sendo que só são válidos para tais ações utilizadores que existam dentro do grupo e utilizadores existentes no sistema, para cada um dos casos respetivamente.
- **Permissões de Pastas de Grupos**: As pastas dos grupos têm permissões definidas para permitir acesso apenas aos membros do grupo. Para além disso, apenas os membros pertencentes a um grupo conseguem aceder à lista de membros desse mesmo grupo.

4 Conclusão

Em suma, o projeto foi desenvolvido com foco na segurança e modularidade. As funcionalidades essenciais foram implementadas, permitindo uma comunicação eficiente e segura entre os utilizadores locais. No entanto, há espaço para melhorias futuras, como a integração com serviços externos e a adição de funcionalidades avançadas de segurança.

Melhorias Futuras:

- **Envelope para mensagens:** As mensagens no projeto desenvolvido são guardadas com os seus metadados no mesmo ficheiro, isto pode facilitar possíveis injeções. Sendo que para resolver este problema poderíamos criar um envelope que guardasse estes metadados separados das mensagens.
- **Acesso à queue**: Quanto à "Mail Queue" temos um utilizador com acesso total a esta e que é utilizado por todos processos que precisem de manipular esta. Sendo que a melhoria seria criar vários utilizadores com permissões diferentes, de acordo com as exigências dos processos. De forma a que caso um processo seja comprometido não tem acesso total à "queue".
- **Criação de mais processos:** De forma a melhorar a segurança, achamos que seria relevante adicionar mais processos para compatibilizar os acessos. Dois exemplos seriam um processo para limpar as mensagens já tratadas da "Mail Queue" e outro que executaria como "local user" que guardava as suas próprias mensagens.