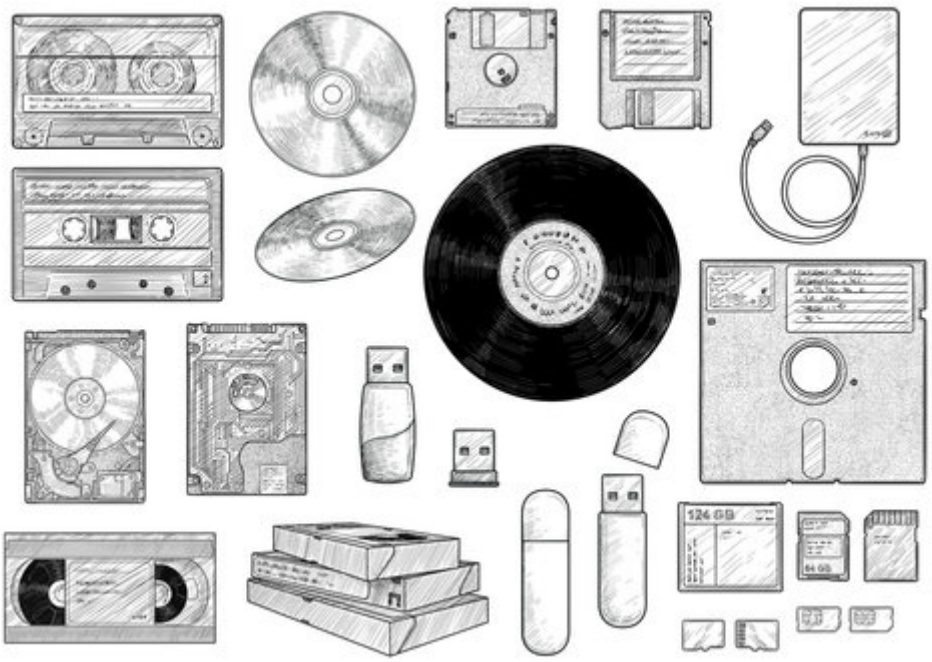
1 2 9 0 FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE D COIMBRA

dei25 departamento de engenharia informática 1995 – 2020

Sistemas Distribuídos 2021/22

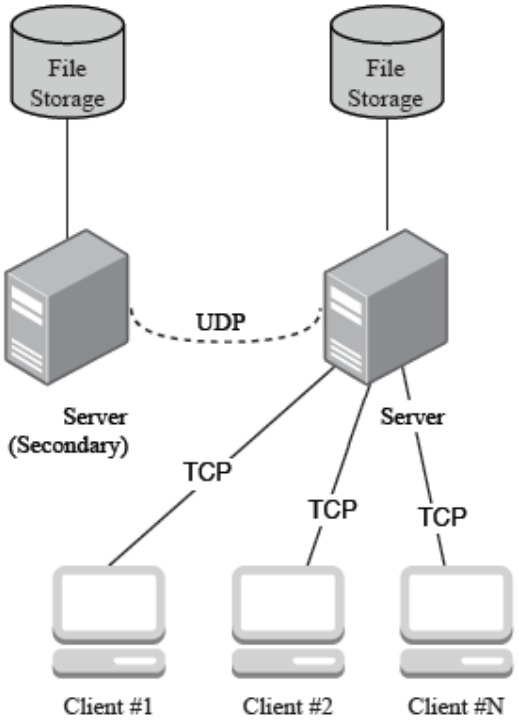
Trabalho prático
“ucDrive”

Meta 1



Ana Beatriz Marques
2018274233, anafm@student.dei.uc.pt
Bárbara Gonçalves
2018295452, bsfg@student.uc.pt

Arquitetura de Software



File Storage: Pasta onde os documentos do servidor estão armazenados, definido no server.properties.

TCP: Protocolo que realiza a conexão entre o servidor e o cliente. É criada uma thread para cada novo cliente. Enviam-se comandos do cliente para o servidor através do socket de comandos, recebendo a resposta do servidor e, caso seja pedido pelo cliente a opção de download ou upload, é criada uma thread que cria um socket para ficheiros e realiza a transferência dos mesmos.

UDP: Protocolo que realiza a conexão entre o servidor primário e secundário. Realiza o heartbeat, e quando se faz um novo upload para o main server, é criada uma thread que abre o socket de ficheiros e envia o file. Simultaneamente envia mensagem pelo heartbeat para criar thread para a leitura do ficheiro no servidor backup, desta forma é sempre feita uma cópia para o backup server.

Server: Realiza os comandos enviados pelo cliente. Tem uma thread "mãe" que vai aceitar clientes e criar uma conexão tcp, sendo que esta vai criar uma thread nova e abrir um socket de comandos para cada cliente.

Client: Interface de cada cliente. É criada uma thread que vai correr a aplicação cliente, onde o utilizador poderá realizar várias ações, tais como mudar entre diretorias, listar ficheiros, alterar a password e fazer download e upload de ficheiros.

Funcionamento do servidor ucDrive

Ao iniciar o servidor é necessário ter os seguintes argumentos:

{ip address} {port}

De seguida, o servidor vai ler o ficheiro "server.properties" e atribuir o respectivo ip e port predefinido. Quando o ip e o port não correspondem a nenhum dos 2 possíveis endereços do servidor, ele ignora os argumentos e identifica-se como uma das possibilidades.

Segue então cronologicamente os seguintes passos:

- criação da thread de update do ficheiro objeto e respectiva leitura
- inicia a ligação UDP: cria um socket e inicia o protocolo de comunicação UDP com outros servidores online
- inicia a ligação TCP: Servidor cria uma thread que abre um serversocket. Este fica à espera que clientes se conectem ao ip e porto do servidor funcional, pré-definidos no ficheiro client.properties. Por cada cliente ligado ao servidor, é criada uma ligação tcp, que cria uma thread que vai lidar com os comandos inseridos pelo cliente, permitindo comunicação entre o servidor e o cliente .

Assim que a aplicação do cliente (terminal) é iniciada, é pedido ao utilizador que faça login. Caso a sua informação corresponda com um dos utilizadores pré-definidos no ficheiro *people.txt,* este é

autenticado e entra no menu principal, onde vai poder realizar várias operações.
O cliente insere na consola o que deseja fazer e esse comando vai ser enviado para o servidor. Sempre que o servidor recebe um comando do cliente através do TCP, este chama a função adequada à sua execução, e devolve uma resposta que vai ser enviada para o cliente através dessa mesma conexão. Esta comunicação é feita com a função auxiliar *writeRead(Socket s)*, que escreve (out.writeUTF(data)) e lê (in.readUTF()) do socket de comandos.

Quando o cliente deseja fazer um download ou upload, é enviado esse comando para o servidor, que vai criar uma nova thread que abre um novo serversocket de transferência de ficheiros. A port deste socket é gerada de forma random (*new ServerSocket(0)*), e é depois enviada ao cliente, para que este possa abrir um socket nesse mesmo porto. Tendo em conta as permissões de navegação cada utilizador, são então realizados os downloads/uploads desejados. Após a finalização de um upload para o servidor, este, através do UDP, copia o ficheiro e envia-o para o backup server.

Mecanismo de Failover

Com a implementação do heartbeat, é possível determinar e alterar o estado do servidor: main ou backup.
Quando um determinado número de pings entre o main e o backup server são perdidos é porque um dos servidores deixou de funcionar e é necessário realizar o failover:
Sempre que o main server falha o backup assume o status main, isto tem impacto nos clientes pois tem de ser reconectado ao novo servidor main.

Como determinamos main/backup com a resposta do primeiro heartbeat (se existe heartbeat é porque já existe outro servidor ativo logo é backup; se o heartbeat gera erro o server é único logo main) garante que sempre que o servidor que falhou voltar a recuperar ele assume o status de backup.

Distribuição de tarefas

Devido a sermos um grupo de 2 pessoas, a distribuição realizada para a primeira meta foi realizada da seguinte forma:

- Ana Beatriz: realização do Servidor e conexões UDP
- Bárbara: realização da Consola Cliente e conexões TCP

Ambas realizamos vários testes.

Testes

Teste	pass/fail	notas
Heartbeat entre servers	pass	sempre que se inicia um servidor ele envia um ping único para determinar o seu status: main ou backup
Server secundário assume quando falha o primário	pass	
Login	pass	
Alterar a password	pass	
Alterar a diretoria remota	pass	
Alterar a diretoria local	pass	
Listar ficheiros remotos	pass	
Listar ficheiros local	pass	
Alterar ip e port	pass?	A alteração é feita automaticamente no failover pois o cliente tem acesso à informação de ambos os servidores, a alteração manual é redundante mas mesmo assim possível
Download	pass	
Upload	pass	
Logout	pass	
Atualização de ficheiros entre servidores	pass	