Trabalho Prático

MSGDIST

META 2

Sistemas Operativos

2019/2020

Daniel Moreira Ribeiro Nº2017013425

Jorge Alberto Martins Coelho Nº2017015323

Índice

Estratégia Geral 3
Estrutura de cada ficheiro 3
Lógica da solução 4
Estruturas 5
Mecanismos de comunicação 5
Descrição das mensagens trocadas entre processos 6

Estratégia Geral

Numa fase inicial do trabalho foram feitas as estruturas pedidas para a realização desta fase.

Na execução do cliente, há receção e armazenamento de um "username". Este só continua a sua execução se o gestor estiver iniciado. O cliente de 10 em 10 segundos verifica a sua conexão com o gestor. No menu de comandos do cliente podemos escolher escrever uma mensagem(que será enviada para o servidor), listar os tópicos existentes, procurar títulos de mensagens, procurar mensagens e desligar o cliente.

Para iniciar o gestor é necessário receber inicialmente um número máximo de mensagens possíveis que este pode armazenar e o numero de palavras proibidas que as mensagens podem ter. Essas variáveis no entanto não estão conectadas com as respetivas tarefas pretendidas. Simultaneamente corre um verificador de que o servidor esta online para o cliente, o atendimento de mensagens do cliente, o atendimento de comandos do cliente e o menu de comandos do Administrador. O administrador pode desligar o servidor, ativar ou desativar o filtro de palavras proibidas e verificar uma palavra.

A criação do "makefile" foi feita seguindo os parâmetros do enunciado.

Existem ainda os ficheiros "cliente.h", "gestor.h" e "estruturas.h" para armazenar certas linhas de código.

Estrutura de cada ficheiro

O cliente.c recebe um argumento para armazenar o "username" que é necessário para a sua execução. Este é um argumento de uma estrutura que foi inicializada no estruturas.h. Este ficheiro possui um menu com múltiplos comandos descritos na "*Estratégia Geral*".

O gestor.c recebe um número correspondente ao número de mensagens que poderá armazenar e o numero de palavras proibidas mínimo para não aceitar uma mensagem através da linha de comandos quando é executado. Simultaneamente correm 3 threads e uma função. Há então simultaneamente a confirmação de que o servidor esta online para o cliente, o atendimento de mensagens do cliente, o atendimento de comandos do cliente e o menu de comandos do Administrador. O servidor recebe uma informação do cliente de 10 em 10 segundos e volta a enviar confirmando assim a sua ligação. No atendimento de mensagens o servidor recebe a mensagem e aloja numa estrutura, numa posição do índice do array da estrutura. Após isso o corpo da mensagem parte em palavras e uma a uma são analisadas com o verificador.c através de pipes enviando palavras e recebendo um valor consoante a existência da palavra no "palavrasProibidas.txt" caso o filtro de palavras proibidas esteja ligado. A variável que define um limite máximo de palavras proibidas foi implementada mas não executa a sua tarefa. O valor da duração da mensagem que foi recebida irá ser argumento de um alarme que irá dar valor nulo ao titulo e ao corpo mantendo assim o tópico e irá incrementar a variável responsável pelo limite máximo de mensagens que não está a exercer na sua tarefa apesar de

possuir um valor introduzido na execução do gestor. O atendimento de comandos do cliente passa por receber e enviar informações através de FIFO tais como as listas de tópicos e títulos. Por fim, os comandos do administrador podem ser utilizados para verificar palavras individuais, desligar e ligar o filtro de palavras proibidas e desligar o servidor.

Os header files do projeto apresentam apenas includes, estruturas e defines que são utilizadas nos dois ficheiros principais.

Lógica da solução

No ficheiro "makefile" existe a possibilidade de serem efetuados oito comandos diferentes.

Comando "all" cria todos os ficheiros executáveis do projeto.

Comando "cliente" cria o executável do cliente.

Comando "gestor" cria o executável do gestor.

Comando "verificador" cria o executável do verificador.

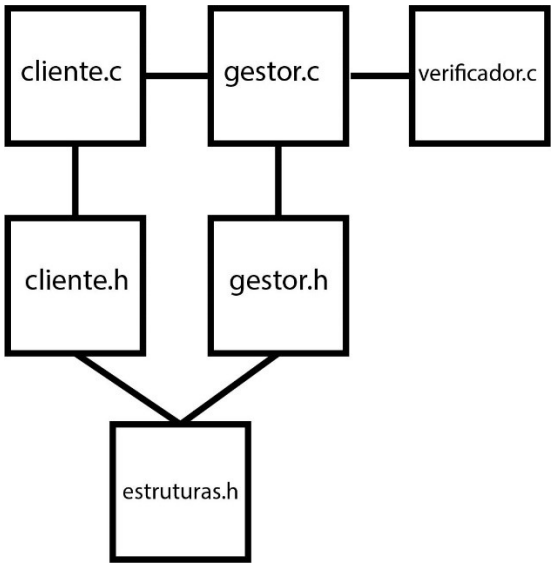
Comando "cliente.o" cria o objeto do cliente.

Comando "gestor.o" cria o objeto do gestor.

Comando "verificador.o" cria o objeto do verificador.

Comando "clean" elimina todos os ficheiros executáveis do projeto.

Associados aos comandos correspondentes do seu ficheiro ".c" estão ainda associados os header files que têm influência no desempenho de cada ficheiro.



Estruturas

A estrutura MENSAGEM é usada para o armazenamento dos vários campos que uma mensagem possivelmente terá, sendo estes o ID, o tópico, o título, o corpo e a duração das mesmas.

Quanto à estrutura UTILIZADOR o seu propósito será o armazenamento do ID do cliente, o seu "username" e as subscrições que o mesmo efetuará.

A estrutura ORDEM destina-se aos pedidos feitos pelo cliente ao gestor como a procura de um titulo ou de um tópico, alojando o pid do cliente.

A estrutura RESPOSTAGESTOR aloja a resposta dada pelo gestor ao cliente após os seus pedidos.

A estrutura PERGUNTAGESTOR destina-se à confirmação da ligação entre cliente e gestor por parte do cliente.

Mecanismos de comunicação

No cliente.c os mecanismos de comunicação são bastante simples, através da função "getopt" recebe-se através da linha de comandos um "username" que é armazenado numa estrutura própria.

O gestor.c utiliza a forma mencionada anteriormente no cliente.c para receber um número máximo de mensagens que será atribuído à variável MAXMSG e o numero de palavras

proibidas que será armazenado em MAXNOT. A comunicação entre servidor e gestor é feita entre 3 named pipes. Destinam-se cada um deles ao envio de mensagens e resposta, à confirmação do servidor online e ao pedido de ordens tais como listar topicos. O comando "shutdown" desliga o gestor através de um sinal, e o comando "verifica" vai, através de um fork, criar um processo filho dedicado ao "verificador.c". Isto também acontece ao tratar a mensagem recebida em que o corpo desta passará pelo verificador palavra a palavra. Este verificador tem como função ir a um ficheiro de texto e retornar-nos informação acerca da presença de uma palavra no mesmo.

Descrição das mensagens trocadas entre processos

Existem dois processos que efetuam a troca de mensagens entre si, sendo estes o gestor e o verificador.

Primeiramente é escrita a palavra no gestor, depois são criados dois pipes. É feito um fork. O processo pai vai fechar a parte de leitura do "writepipe" pois não necessita de o ler. Vai fechar também a parte de escrita do "readpipe" porque não necessita de escrever nesse pipe.

De seguida o pai escreve na parte de escrita do "writepipe" e de seguida fecha-o.

O processo filho fecha a parte de leitura do "readpipe" porque não precisa de o ler e depois fecha a parte de escrita do "writepipe" porque não precisa de escrever. Depois associa o valor lido do "writepipe" ao "STDIN" do verifica e o "STDOUT" do "verifica" à parte de escrita do "readpipe".

Fecha a parte de leitura do "writepipe" pois não precisa de ler e fecha a parte de escrita do "readpipe" porque não precisa de escrever.

Faz o "execl" do verificador.

É sabido que o "execl" só retorna em caso de erro, por isso, voltando ao processo pai, irá ser feita a leitura do "readpipe" que foi associada ao valor do "STDOUT" sendo alojada numa variável inteira.

Por último, fecha-se a parte de leitura do "readpipe".