



UNIVERSIDADE D COIMBRA

Sistemas informáticos - SI

Francisco Relvão
2018285965|MIEF

Gonçalo Gouveia
2018277419|MIEF

10 de maio de 2021

TCP Protocol

No protocolo de comunicação, existem dois intervenientes, o cliente e o servidor, que comunicam de acordo com um protocolo baseado em texto.

O protocolo usado é *Transmission Control Protocol* - TCP - durante uma conexão tem 3 fases fundamentais: estabelecimento da ligação, comunicação e fecho de ligação. O estabelecimento da ligação é feito em três passos, enquanto que o fecho é feito em quatro. Durante a inicialização são inicializados alguns parâmetros, como o Sequence Number (número de sequência) para garantir a entrega ordenada e robustez durante a transferência.

1 Client

O cliente é a interface gráfica que permite a comunicação entre o user e o server. O objetivo é que este cliente seja uma *thin client*, e, portanto, qualquer tipo de verificação de inputs, jogadas permitidas e controlo do fluxo de execução do programa não fazem parte do domínio do cliente. O cliente é essencialmente um minion do servidor. O cliente só tem iniciativa própria unicamente para unicamente conectar ao servidor. Após a conexão, o controlo do programa passa para o servidor.

O cliente nunca recebe texto com frase ou texto que representem o jogo. Na verdade, a comunicação entre o cliente e o servidor é feito por comandos de extensão relativamente pequena, no máximo 3 palavras, sem espaços e em letras maiúsculas. O cliente limita-se a ler esse comando. De seguida, a ordem é levada a cabo e o utilizador é informado de forma gráfica.

1.1 Conexão com o servidor

Inicialmente, o cliente pergunta ao utilizador se quer iniciar a socket com o servidor. Em caso positivo "Y", a socket é inicializada, a comunicação entre o cliente e o servidor é iniciada. Caso contrário, o cliente termina a sua execução "N". Caso o cliente não tenha inserido uma palavra valida é pedido ao utilizador para repetir a escolha.

1.2 Credenciais

- username - user
- password - 123

De seguida, o cliente pede as credencias ao utilizador. É imperativo que as credenciais sejam exatamente iguais às que estão armazenadas no servidor. Espaços no nome do utilizador não são ignorados. Se as credenciais de login estiverem corretas, recebe ("VALID_CREDENTIAL"), o servidor avança para inicialização do jogo. Se o utilizador introduzir credenciais erradas, o cliente recebe do servidor ("INVALID_CREDENTIAL"). O cliente termina a sua execução, se o utilizador recusar recolocar as credenciais, enviando um comando ("CLOSE_CLIENT") ao servidor.

1.3 Número de discos

Após a validação das credenciais, é pedido o número de discos (entre 3 e 10). O número inserido pelo utilizador na consola do cliente é verificado pelo servidor. Caso seja válido, o cliente recebe um comando (DISKS_ACCEPTED) e o jogo continua, caso contrário, o servido envia um comando ("DISKS_DECLINED") que pede ao cliente novamente um número entre 3 e 10. Caso o cliente tenha inserido um formato inválido então o cliente recebe um comando ("DISKS_DECLINED_ERROR") que avisa o cliente do erro e pede novamente o número de discos (entre 3 e 10).

1.4 Pino inicial e final

É pedido ao utilizador para escolher um pino inicial e um pino final (A, B ou C). Se as opções enviadas pelo cliente para o servidor forem validadas então o cliente recebe um comando ("PIN_SELECTED") e o programa continua, caso as opções do cliente não sejam validadas o cliente recebe um comando ("PIN_ERROR") e é pedido ao cliente novas opções. Se as opções enviadas pelo cliente não tiverem um formato válido por exemplo uma *String* então o cliente recebe um comando ("PIN_INVALID_NUMBER").

1.5 Movimentos

O cliente pede um movimento (de 1 a 6) ao utilizador e transmite a resposta ao servidor. Após o servidor verificar a resposta e enviar o comando ("DISKS_ACCEPTED") o cliente faz a jogada pedida, que mostra o tabuleiro de jogo ao utilizador, para isto o server envia um comando de ("DRAW"), também é incrementado o numero de rondas com o comando ("COUNTER_ADD"). Este passo repete-se até o utilizador ganhar ou desistir. Caso o servidor rejeite a opção do cliente, o cliente recebe o comando ("DISKS_DECLINED"), se a opção enviada pelo cliente não tiver um formato valido por exemplo uma *String* então o cliente recebe um comando ("DISKS_ERROR"). Caso o utilizador queira sair do jogo pode inserir a opção "Y", enviando o código ao servido ("Y").

1.6 Menu final

Quando o jogo termina, o servidor comunica ao cliente que o jogo acabou com o comando ("WIN") e abre o menu no cliente com o comando ("MENU").o cliente tems as seguintes opções:

1. Play Again - Caso o cliente opte por fazer um novo jogo, esta opção é comunicada ao servidor ("OPTION1"), do lado do servidor sair do ciclo de jogo, e recomeça um novo jogo pedindo todas as novas opções de jogo e inicializando todas as variáveis de jogo ("PIN_CLEAR") retirando os discos do pino final e (COUNTER_RESET). Do lado do cliente recomeça um novo jogo com novas opções de jogo.
2. Estáticas - Caso o cliente opte por esta opção,esta opção é comunicada ao servidor ("OPTION2"). Acessa a base de dados de estatísticas do cliente, que mostra ao utilizador as suas estatísticas.
3. Quit - Caso o cliente opte por terminar a sessão o cliente é desconectado do servidor, enviando a opção numero 3 do Menu final("OPTIONQ"), o servidor comunica que recebeu essa informação, enviando ao cliente ("CLOSE_CLIENT").

2 Server

O servidor nunca envia ao cliente texto com frases ou texto que representem o jogo. Na verdade, a comunicação entre o cliente e o servidor é feito por comandos de extensão relativamente pequena e com instruções, no máximo 3 palavras, sem espaços e em letras maiúsculas. O cliente limita-se a ler esse comando e executar procedimentos simples.

2.1 Conexão com o cliente

O server fica a escuta a espera que um cliente se queira conectar ao mesmo. Caso um cliente aceite se conectar ao servidor, a Socket é inicializada no lado do cliente. Seguidamente o servidor envia a mensagem ao cliente ("LOG_IN"). Caso o cliente receba um comando "N", este é desconectado. Qualquer outra opção do cliente o servidor envia um comando ("INVALID_COMAND") para pedir uma nova entrada do cliente.

2.2 Credenciais

O servidor recebe o username e a password do cliente e a verifica se estas credenciais fazem parte da base de dados. Se as credenciais forem corretas o servidor envia a mensagem ("VALID_CREDENTIAL"), caso as credenciais estejam erradas o servidor envia a mensagem ("INVALID_CREDENTIAL") ao cliente.

Quando as credenciais estão erradas o servidor também envia o código ("CLOSE_CLIENT") que faz o cliente se desconectar do servidor.

2.3 Número de discos

O servidor recebe a opção enviada pelo cliente e verifica se opção é permitida. Caso a opção inserida pelo cliente seja valida o servidor comunica com o cliente que a opção é valida ("DISKS_ACCEPTED"). Se a opção inserida pelo cliente for invalida o servidor comunica ao cliente ("DISKS_DECLINED") e seguidamente envia para inserir o numero de discos de jogo valido com o comando ("DISK_NUMBER"). Quando o cliente envia uma opção que não tenha um formato valido (ex. uma *String*) então avisa o cliente com um comando ("DISKS_DECLINED_ERROR") e envia seguidamente um comando ("DISK_NUMBER") para o cliente inserir um numero de discos de jogo valido.

2.4 Pino inicial e final

O servidor recebe a opção inserida pelo cliente, em uma primeira analise para o pino inicial, avalia se a letra inserida corresponde a 1, 2 ou 3. Posteriormente, o servidor analisa se o pino final é diferente do pino inicial. Se o server validar a resposta envia a mensagem ("PIN_SELECTED") ao cliente. Após esta verificação, o jogo pode continuar. Se um dos dados for invalido, é enviado o código ("PIN_ERROR") e pede novamente ao cliente um pino inicial e outro final.

2.5 Movimentos

O servidor pede ao cliente que movimento de disco que fazer. Consoante a resposta do cliente, o servidor avalia a resposta, se a opção é valida. O servidor envia ao cliente que é possível fazer a jogada. O counter é incrementado, ("COUNTER_ADD"). O cliente que mostra o tabuleiro de jogo ao utilizador. Se o movimento for inválido (não se encontrar entre 1 e 6, inclusive, ou se não for possível jogar), o servidor dá indicação ao cliente para voltar a pedir um movimento ("MOVE_ERROR").

Caso o servidor receba o comando ("Y"), o jogo acaba.

Em qualquer outra opção recebida pelo o servidor, o cliente recebe um código ("DEFAULT") a pedir uma opção valida. No final de um movimento, é verificado o tamanho da torre escolhida como torre final. Se por ventura, o tamanho da torre escolhida equivale ao número de peças mais a base na stack, então o jogo termina, é indicado para que o cliente apresente o número de passos mínimos e os passos efetuados. Por fim, o menu é apresentado.

Naturalmente, se o tamanho da torre não for o indicado anteriormente, então o jogo continua para a próxima ronda.

2.6 MENU

Quando o jogo acaba, o servidor envia a mensagem de abrir o menu ao cliente ("MENU"). Recebe a opção seleccionada pelo cliente:

1. Novo Jogo - O servidor recebe a opção do cliente ("OPTION1") e dá reset nas variáveis de jogo. Recomeça um novo ciclo de jogo.
2. Estatística - O servidor recebe a opção do cliente ("OPTION2"). O servidor volta ao menu.
3. Quit - O servidor recebe a opção do cliente ("OPTIONQ"). O cliente é desconectado do servidor e este fica à espera de um novo cliente.

Qualquer outra opção inválida inserida pelo utilizador, o servidor envia um comando default ("OPTIOND") a pedir ao cliente uma nova opção valida.

3 Diagrama de funcionamento

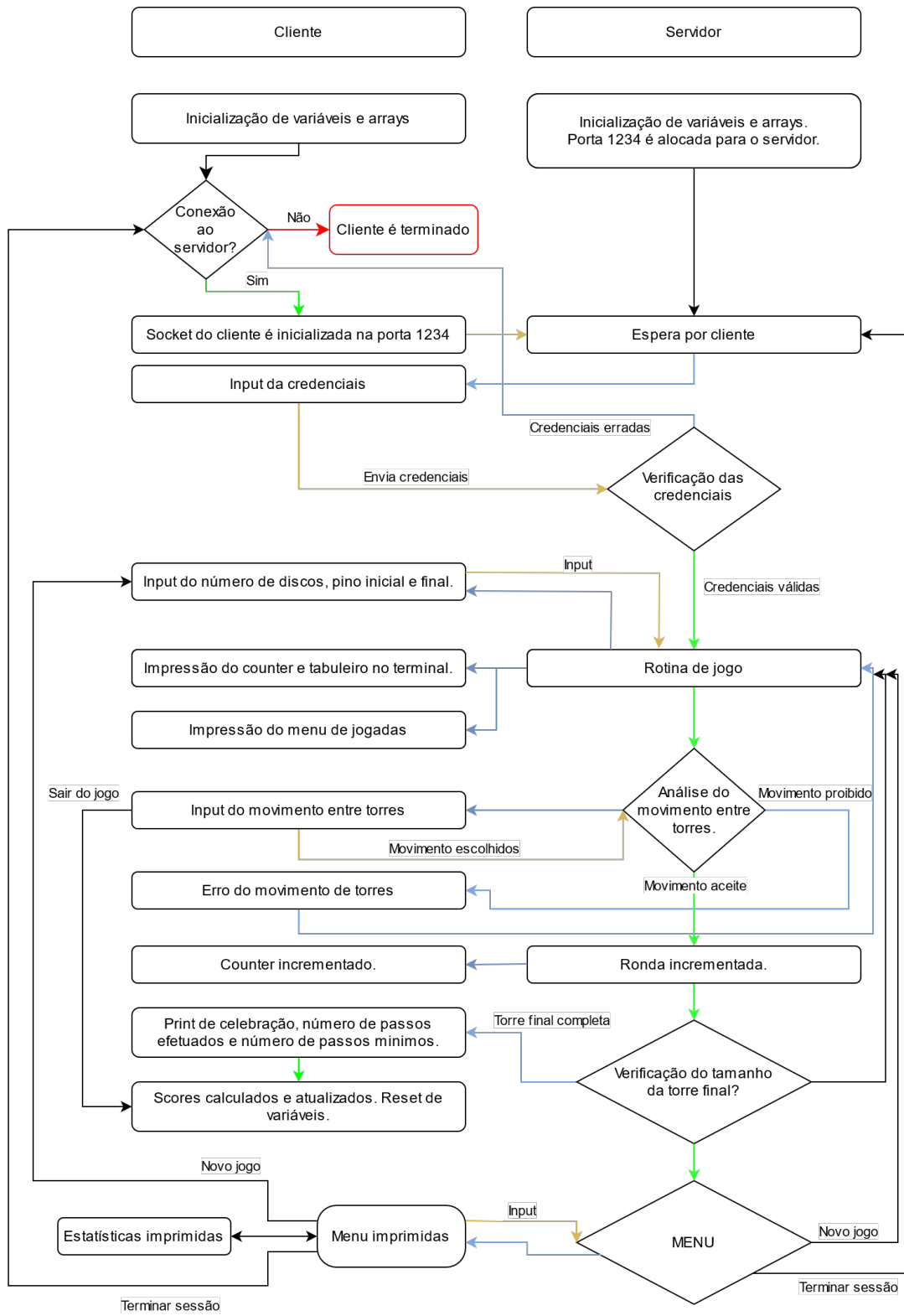


Figura 1: Esquema do funcionamento e interação entre o servidor e o cliente.