

**Universidade Federal
de São João del-Rei**

Departamento do Curso de Ciência da Computação
Programa de Graduação

Documentação do Primeiro Trabalho Prático de Redes

Alunos: Jardel Carvalho, Millas Násser
Professor: Rafael Sachetto

Outubro
2018

Conteúdo

1	Introdução	1
2	Contexto	2
2.1	Programação em Sockets	2
2.1.1	História	2
2.1.2	Funcionamento	2
2.1.3	Unix domain socket	2
2.2	Truco	2
3	Materiais e Métodos	3
3.1	Linguagem C	3
3.2	Protocolo TCP para a Camada de Transporte	3
3.3	Regras do jogo de Truco	4
4	Estratégia Proposta	6
4.1	Comunicação entre Cliente e Servidor	6
4.2	Diretrizes para uma comunicação consistente	6
5	Conclusão	7

1 Introdução

Desde tempos remotos é percebido que o entretenimento é uma necessidade básica do ser humano, uma prova disso são as diversas descobertas de instrumentos e lugares ao redor do mundo, citando alguns exemplos destaca-se flautas com idades aproximadas de 42 à 43 mil anos [3], o Coliseu localizado na cidade de Roma e um jogo de tabuleiro datado de mais de 3000 A.C., o Senet [4].

Uma classe de jogos bastante conhecida por todo o mundo são jogos que envolvem cartas, caracterizando algum tipo de baralho. Sua primeira aparição é datada do século 9 durante a dinastia Tang, na china [5]. Sendo caracterizado como primeiro jogo do mundo o chamado “Leaf Game”, foi jogado pelo clã Wei (clã do marido da princesa Tongchang). Com o passar dos anos nasceram diversos outros jogos ao redor do mundo, não se restringindo a apenas à China. Devido a grandiosidade dos jogos de baralho em alguns momentos eles foram caracterizados como algo místico, caracterizando o surgimento dos baralhos com premissas de previsão de acontecimentos futuros, tal como o Tarot [2].

Estes podem ser jogos individuais, como por exemplo, paciência. Ou coletivos, variando consideravelmente o número de pessoas envolvidas, um exemplo é o jogo de póquer, que pode variar de 2 à indefinidas pessoas jogando uma mesma partida. Por serem jogos muito populares, eles sofrem diversas variações em suas regras, tal como a quantidade de cartas em uma mão, a disposição delas na mesa e até mesmo na forma de vencer cada rodada, mas em ambos os casos é utilizado o mesmo conjunto de cartas (baralho).

Outro jogo bastante popular, principalmente em vários lugares da América do Sul e em alguns lugares da Espanha e Itália, é o Truco [1]. Com regras peculiares, o jogo possui variações para ser jogado entre 2, 4, 6, 8, 10 e 12 pessoas, sempre divididos em 2 equipes. Nas variações de 6 à 12 as rodadas ímpares são jogadas pela equipe e nas pares jogadas individualmente por cada um.

O trabalho consiste em fazer um jogo de truco em rede, que por questões de simplificação, optou-se por escolher utilizar apenas as variações de 2 e 4 pessoas. Nesses casos, cada jogador faz a sua jogada e os demais aguardam a sua hora de jogar. Para a implementação foi descrito uma série de restrições, tal como, a transmissão de dados deverá ser feita entre o cliente e o servidor, nunca entre clientes, utilizando uma API fornecida pelo sistema operacional (Sockets). Tal como o jogo deverá ser implementado de forma que o servidor do jogo, seja feito em Unix juntamente com todas as consistências e regras. E por fim o cliente deverá ser somente uma interface com os jogadores.

2 Contexto

2.1 Programação em Sockets

2.1.1 História

O surgimento da programação em sockets se deu no ano de 1980 nos Estados Unidos a partir do órgão ARPA (Advanced Research Projects Agency of the Department of Defense).

A ARPA, a partir do incentivo a Universidade da Califórnia, contribuiu para que tal instituição de ensino viesse então a desenvolver um sistema operacional que seria então utilizado pela ARPAnet, do qual, é o antecessor a internet dos tempos atuais.

A partir deste fomento, veio a ser desenvolvida uma interface destinada ao sistema operacional Unix BSD (Berkeley Software Distribution), popularmente conhecida como Berkeley Sockets Interface, que veio a se tornar a base para a maior parte das interfaces e protocolos de internet TCP/IP atuais.

2.1.2 Funcionamento

De forma resumida e objetiva pode-se dizer que a programação em sockets fornece uma interface entre aplicações com o devido uso de protocolos disponíveis para utilização na camada de transporte.

2.1.3 Unix domain socket

O Unix domain socket ou soquete de domínio Unix, também conhecido como IPC(Inter-process Communication) é uma forma de comunicação entre processos que executam em máquinas com sistemas operacionais baseados no Unix. Um soquete de domínio Unix fornece recursos para comunicação a partir do uso de datagramas ou segmentos em streaming. Também é possível realizar a comunicação entre processos que compartilham tempo de processamento na mesma máquina, o que o faz ter uma certa semelhança com pipes.

2.2 Truco

Neste trabalho está sendo utilizado a variação do truco para 2 e 4 pessoas, e nas duas as regras são as mesmas. Mas ainda nessas duas variações de quantidades de jogadores, ainda há uma variação entre a quantidade de pontos a serem distribuídos e a ordem das maiores cartas, são conhecidos

pelas principais variações brasileiras do jogo, o Truco Paulista e o Truco Mineiro. As semelhanças entre eles é o próprio conjunto de cartas utilizado o padrão francês de cartas, que se tornou o padrão mundial. Consistindo em quatro naipes e 13 números. Os naipes são Paus, Copas, Espadas e Ouros e os números possuindo 10 valores distintos entre si, representados por valores de 1 a 10, e outros três, Q, J e K simbolizando respectivamente os valores 8, 9 e 10, possuindo precedência sobre os demais valores. Mas diferentes de vários outros jogos de cartas, o Truco[1] não utiliza as cartas 8, 9 e 10.

A principal diferença está na ordem das manilhas (cartas mais valiosas do jogo) que no Truco Mineiro elas são estáticas, chamadas manilhas velhas. Enquanto no Truco Paulista as manilhas são definidas de acordo com a carta no topo do baralho, chamada de vira (sendo a mesma descartada logo após), sendo as manilhas chamadas de móveis. Outra diferença importante a ser destacada é na distribuição de pontos onde na variação paulista o tento (ponto) vale apenas 1 unidade, já na versão mineira aposta-se 2 tentos. Para este trabalho foi considerado o Truco Mineiro durante a sua implementação.

3 Materiais e Métodos

3.1 Linguagem C

Para este trabalho foi utilizada a linguagem C, proposta por Dennis Ritchie em 1972 através da Bell Labs, destinada ao desenvolvimento do sistema operacional Unix. A linguagem C conta com a utilização do paradigma imperativo do qual é fortemente baseado na arquitetura de Von Neumann que consiste na troca de informações entre memória principal, memória secundária, centro de processamento, unidade de controle e mecanismos de entrada e saída.

3.2 Protocolo TCP para a Camada de Transporte

Para a camada de transporte, foi utilizado na aplicação, o protocolo TCP (Transmission Control Protocol). O protocolo TCP é um dos protocolos da camada de transporte que são predominantes na Internet. Seu diferencial se encontra na capacidade de transferir dados de maneira segura e consistente. Suas principais características podem ser definidas conforme a seguir:

- Orientado à conexão: A aplicação solicita a conexão, a partir disto, é instaurada uma via para transferência de dados entre pares.

- Handshake (Aperto de mãos): Sequência lógica de passos para garantir que a autenticação e o envio de pacotes seja feita de maneira completa e segura.
- Ponto a ponto: Comunicação entre pares garantida pelos IP's de remetente e destinatário presentes nos cabeçalho do pacote.
- Full duplex: É possível transmitir informações a partir dos dois lados da conexão simultaneamente.
- Entrega ordenada: A transferência dos pacotes no protocolo TCP pode ser fragmentada, com isso, a ordem de chegada dos fragmentos pode não seguir uma ordem desejada, por isso o próprio protocolo TCP tem recursos para garantir que a ordenação seja feita durante a reconstrução do pacote.
- Controle de fluxo e congestionamento: O TCP usa o campo janela do cabeçalho para controlar o fluxo de dados. O destinatário, à medida que recebe os dados, envia mensagens ACK (Acknowledgement), constatando o recebimento de um fragmento e informando ao remetente qual o tamanho ideal de fragmento para manutenção do fluxo.

3.3 Regras do jogo de Truco

Como dito anteriormente, o presente trabalho só caracteriza as variações entre 2 e 4 jogadores utilizando manilhas velhas (Truco Mineiro), em um jogo de quatro jogadores formam-se 2 equipes. Alternando entre si a rodada de embaralhamento e distribuição de cartas. Primeiramente é necessário definir qual é a hierarquia das cartas. No Truco existe uma categoria especial para as cartas mais importantes do jogo, conhecidas como manilhas as cartas se caracterizam em importância na seguinte ordem:

- **Zap:** 4 de Paus.
- **7 de Copas**
- **Espadilha:** Às de Espadas.
- **7 de Ouros**

As outras cartas são caracterizadas pela importância seguindo o seu valor, não importando mais o naipe da carta. Ignorando as cartas já declaradas como manilhas, a ordem para as outras cartas se dá por 3, 2, Às, K, J, Q, 7, 6, 5, 4, nessa ordem de grandeza.

Primeiramente é escolhido uma pessoa a embaralhar o baralho, e o adversário à esquerda de quem está embaralhando realiza o corte no baralho. Após esses passos, é realizada a distribuição das cartas para os jogadores, partindo de quem embaralhou, seguindo distribuindo cartas à sua direita até chegar a quem havia feito o corte no baralho. No truco cada jogador recebe 3 cartas por mão, e joga-se alternadamente entre os jogadores.

Quando cartas de valores diferentes são jogadas à mesa vence o jogador que lançou a carta de maior valor segundo a ordem dita. o time que ganha duas ou mais jogadas vence a rodada, ganhando 2 tentos. Em caso de empate o time que ganhou a primeira jogada, vence. Caso seja a primeira rodada, o time que vencer a a próxima rodada vence o jogo. Caso a segunda também empatar, cabe à terceira jogada o desempate. Se nas três jogadas houverem empate, diz se que o jogo “cangou”, e nenhuma equipe recebe os tentos.

Uma situação frequente no Truco é pedir o aumento dos tentos, por meio do “truco” o que caracteriza o nome do jogo, a ordem das apostas é dado pela seguinte ordem:

- **Pedir truco:** o jogo passa a valer 4 tentos.
- **Pedir seis:** o jogo passa a valer 8 tentos.
- **Pedir nove:** o jogo passa a valer 10 tentos.
- **Pedir doze:** o jogo passa a valer 12 tentos.

Por exemplo, a equipe que recebeu um truco, pode pedir seis. Dessa forma o aumento das apostas é feito de forma alternada entre os oponentes. Ao aceitar, o jogo passa a valer mais tentos, de acordo com o pedido de aposta atual. Recusando uma aposta, a rodada é automaticamente finalizada e quem fez a intimação ganha a quantidade de tentos da aposta anterior. Por exemplo, uma equipe pede seis, mas o oponente a recusa, portanto pela regra a equipe que pediu seis irá ganhar 4 tentos, que é a pontuação do *Pedir truco*. Para quem recusar um truco, o time que “trucou” irá ganhar 2 tentos.

Há outra situação especial no Truco, que é quando uma equipe atinge 10 tentos. Na Mão de 10 para uma só equipe, é permitido olhar as cartas do parceiro (isso combinado antes de iniciar o jogo). Caso a dupla resolva jogar, a mão valerá 4 tentos. Caso resolva fugir, os adversários ganham 2 tentos. É proibido trucar, quem trucar em Mão de 10 perderá 4 tentos. Com cartas a provar vitória garantida em Mão de 10, o desenvolvimento da jogada se torna desnecessário, bastando tão somente que essas cartas sejam mostradas. Havendo empate, ninguém ganha tento, passando-se o monte adiante para ser embaralhado. O jogo é ganho pela primeira equipe que conseguir acumular 12 tentos em uma partida.

4 Estratégia Proposta

4.1 Comunicação entre Cliente e Servidor

Para o desenvolvimento do jogo de truco foi adotado o modelo de comunicação Cliente/Servidor. O modelo Cliente/Servidor propõe que a comunicação entre os Clientes seja mediada por um Servidor, ou seja, se um determinado Cliente deseja se comunicar com outro Cliente, o mesmo deve por meio de troca de informações com o Servidor tornar essa tarefa concluída.

Entretanto, apesar da nomenclatura do modelo, pode-se constatar o fato de que o Cliente se comporta em determinados instantes como um Servidor, assim como o Servidor também pode se comportar como um Cliente.

Para o desenvolvimento do jogo de Truco proposto no trabalho, foi adotada uma padronização de troca de informações entre ambos os tipos de Hosts.

O Cliente atua na leitura de novas informações a partir de uma porta fornecida pelo Servidor, por sua vez, o Servidor, envia segmentos para esta porta, com o pretexto de informar ao Cliente quais são as opções que o mesmo pode selecionar para que seja dada a interação com os outros jogadores. Um jogador que por sua vez é um Cliente, envia a opção selecionada para o Servidor, do qual, realiza o processamento da jogada e muda o turno para o próximo jogador atuar no jogo.

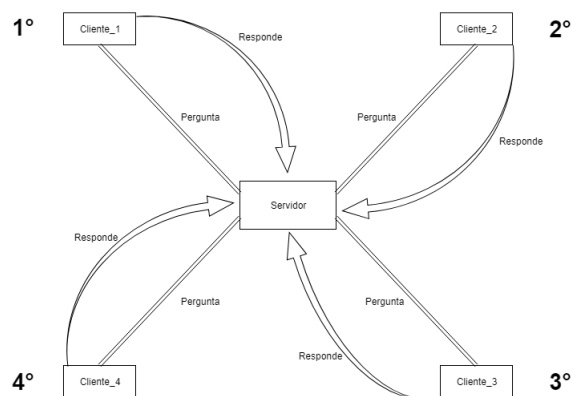


Figura 1: Ciclo de comunicação para o jogo de Truco.

4.2 Diretrizes para uma comunicação consistente

Para padronizar a comunicação entre o Cliente e o Servidor foi definido o tamanho do segmento para 1024 Bytes (1 KB). O tamanho se deve ao fato de

ser necessário o envio informações que correspondem ao nome dos jogadores que interagem no jogo, cartas presentes na mão do jogador assim como outras mensagens de caráter informativo. A partir desta medida padronizada de

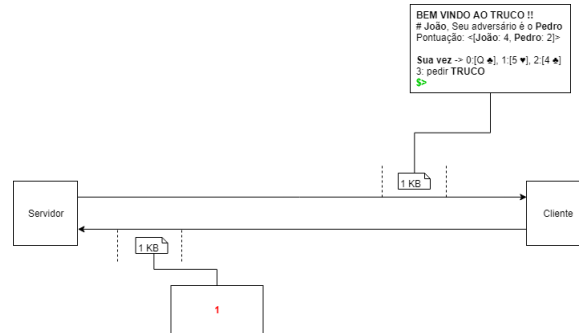


Figura 2: Formato das mensagens.

segmentos, pode-se constatar que um servidor somente pode enviar e ler informações a partir de frações de 1 KB, o mesmo vale para os clientes.



Figura 3: Representação do compartilhamento de mensagens com segmentos de tamanho fixo.

5 Conclusão

No desenvolvimento do trabalho foi constatada grande complexidade na comunicação entre os jogadores. Para meios de simplificação e unificação dos critérios de interação entre os clientes foi delimitado o tamanho do segmento a 1KB. A partir desta diretiva foi possível garantir mais confiabilidade no desenvolvimento do jogo de Truco que funcionaria por meio do modelo Cliente/Servidor. O processamento das operações demandadas pelo Cliente, ou seja, as respectivas jogadas, foram executadas por completo do lado do Servidor, visto que, a necessidade de processamento era extremamente baixa. Assim, foi possível desenvolver uma aplicação para o lado do Cliente que trabalhasse no total desconhecimento dos tipos de dados e funções que correspondessem ao próprio jogo de Truco, portanto, a partir disso, o Cliente só necessitaria de recursos que fornecessem a possibilidade de processar segmentos puros e a partir destes enviar para o servidor uma entrada do jogador,

ou seja, outro segmento. Este protocolo de comunicação entre os jogadores forneceu também grande simplicidade no desenvolvimento da aplicação para o lado do Cliente.

Foram estes dois critérios para comunicação entre os jogadores que garantiram com que a aplicação fosse desenvolvida da forma mais confiável e simples possível, entretanto, vale ressaltar que em sistemas que contam com uma demanda de processamento mais elevada, a concentração da maior parte das tarefas deve ser retida nos Clientes, para assim poder garantir com que o Servidor conte com a menor carga de processamento possível.

Referências

- [1] Nike Arts. *Enciclopedia de los juegos de cartas*, page 111. Indispensables (Barcelona). RobinBook, 1999.
- [2] M.A.E. Dummett and S. Mann. *The game of Tarot: from Ferrara to Salt Lake City*. Duckworth Publishers, 1980.
- [3] Thomas Higham, Laura Basell, Roger Jacobi, Rachel Wood, Christopher Bronk Ramsey, and Nicholas J. Conard. Testing models for the beginnings of the aurignacian and the advent of figurative art and music: The radiocarbon chronology of geißenklösterle. *Journal of Human Evolution*, 62(6):664 – 676, 2012.
- [4] Michael Sebbane. Board games from canaan in the early and intermediate bronze ages and the origin of the egyptian senet game. *Tel Aviv*, 28(2):213–230, 2001.
- [5] R.K.G. Temple. *The Genius of China: 3,000 Years of Science, Discovery, and Invention*, page 130–131. Prion, 1998.