Relatório trabalho II:
Jogo em uma rede em anel

Redes de Computadores I - CI1058
Departamento de Informática
Universidade Federal do Paraná - UFPR
Eduardo Gabriel Kenzo Tanaka - GRR20211791
Vinícius Yuji Hara - GRR20211763
Curitiba - PR - Brasil

I. Descrição

O objetivo do trabalho é implementar o jogo chamado "foda-se" ou "fodinha" em uma rede em anel com 4 máquinas em que o controle do acesso a rede é feito com passagem de bastão. Para realizar a rede é utilizado o Socket DGRAM.

II. Implementação

1. Detalhes da aplicação

No trabalho foi utilizado a linguagem python. No arquivo shared.py cada elemento da lista PLAYERS_ADDR representa uma máquina. Para rodar o programa, cada máquina deve usar o comando "python3 game.py < índice da máquina >", exemplo: A máquina 3 utiliza o comando "python3 game.py 3". A máquina 1 começa com o bastão, ou seja, é o primeiro carteador. Para iniciar o jogo, deve garantir que todas as máquinas estejam conectadas e para iniciar o jogo a máquina 1 deve digitar a letra 'Y'.

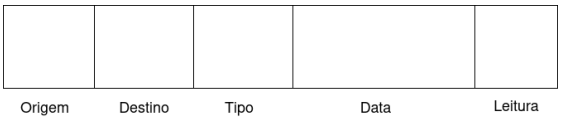
Os jogadores começam com 1 carta e vai até o máximo de cartas que o baralho suporta. O jogo termina caso sobre 1 jogador. Se todos forem eliminados, quem tem mais vida na última rodada é o vencedor, caso o número de vida iguale quando os jogadores forem eliminados, resulta em empate.

Caso um jogador seja eliminado, ele apenas passa a mensagem para o próximo, só irá receber quando é anunciado o vencedor.

Para representar os jogadores foi criado um classe do tipo Player e a classe da mensagem do tipo Message.

2. Estrutura da mensagem

A mensagem tem a seguinte estrutura:



Onde o campo de origem e o campo destino contém o endereço das máquinas. Campo tipo é o tipo da mensagem, campo data o conteúdo da mensagem e o campo leitura ver se destino recebeu a mensagem. Para mensagens onde todos recebem, o destino é 0 e para cada player que recebeu adiciona 1 no campo de leitura.

3. Tipos de mensagem e funcionamento do jogo

Neste tópico irá ser explicado os tipos da mensagens encontrado em shared.py, que é enviado pelo jogador que tem o bastão(carteador) vai enviar para os outros. Todas as funções descritas estão na classe Player.

Na função deal_cards() o carteador envia as cartas com uma mensagem para cada jogador do tipo DEAL. Depois, o carteador informa o vira da rodada enviando uma única mensagem do tipo REVEAL_VIRA para todos os jogadores.

Para coletar os palpites na função collect_guesses(), o carteador envia uma mensagem do tipo GUESS para cada jogador, depois que um jogador dá o palpite, o carteador manda uma mensagem para todos do tipo REVEAL_GUESS informando o palpite do jogador. O carteador é o último a palpitar.

As jogadas das cartas estão na fução card_play começando pelo jogador ao lado do carteador. O carteador manda uma mensagem do tipo PLAY_CARD para cada jogador jogar a sua carta, depois que o carteador recebe uma carta, ele manda uma mensagem do tipo REVEAL_CARD que todos os jogadores escutam e sabem a carta escolhida por um jogador.

A função resolve_round() trata para saber quem foi o vencedor ou empate das cartas jogadas o carteador guarda os pontos dos jogadores.

Depois da mão dos jogadores estarem vazia, o carteador faz o cálculo dos pontos e na mensagem do tipo REVEAL_SCORE, todos os jogadores recebem a mensagem e sabem quais jogadores perderam pontos ou não.

Na função resolve_status(), o carteador manda uma mensagem do tipo COLLECT_STATUS em que todos recebem a mensagem, e se um jogador não tem mais vidas, ele escreve no campo data que foi eliminado. Após o carteador receber a mensagem, se há apenas um jogador restante ou empate, informa todos os jogadores em uma única mensagem do tipo MATCH_WINNER, em que todos escutam a mensagem e acaba o jogo. Se o carteador receber a mensagem e ainda existirem jogadores com vida, manda uma mensagem do tipo REVEAL_STATUS, em que diz quem foi eliminado.

A passagem do bastão é feita pela função pass_token() com uma mensagem do tipo PASS_TOKEN, com o destino sendo o próximo jogador não eliminado.