

Trabalho Prático 1
Redes de Computadores - TN
Iury Alves Bicalho - 2020027016
Abril de 2024 - UFMG

Instruções de compilação

O código fonte foi feito em C++, portanto é necessário compilá-lo com o uso do compilador GCC (GNU Compiler Collection). O arquivo enviado contém todo diretório usado em desenvolvimento, incluindo um Makefile que irá facilitar a compilação do programa principal.

Passos:

1. Abra sua um terminal Linux;
2. Certifique-se que o GCC e Make estão devidamente instalados na sua distribuição do Linux;
3. Desloque-se para o diretório raiz desse trabalho, onde se encontra o Makefile;
4. Utilize o comando sudo apt-get install libjson-c-dev no terminal para instalar a biblioteca <json-c/json.h> requerida para compilação do trabalho;
5. Utilize o comando make no terminal.

Após seguir os passos citados, o executável compilado será criado na pasta `bin`, e poderá ser executado com os seguintes comandos (caso se desloque para a pasta onde está o executável):

- I. `./main <host> <port> itr <id> <nounce>`
- II. `./main <host> <port> itv <sas>`
- III. `./main <host> <port> gtr <N> <sas-1> ... <sas-n>`
- IV. `./main <host> <port> gtv <gas>`

Com <host> podendo ser um endereço IPv4 ou IPv6 ou o nome do servidor.

Estratégia de jogo e reconstrução do estado do jogo

Para simplificar o andar do jogo, o código armazena apenas o ID de cada navio em um vetor, sem se importar com o tipo ou o número de hits que cada um já recebeu. A decisão de atirar também é bastante simples: um loop percorre todos os possíveis canhões no grid rio x ponte, e aqueles que existirem (foram enviados pelo servidor em resposta ao 'getcannons') poderão atirar no primeiro navio da lista de navios dos rios adjacentes e naquela ponte (se as listas estiverem vazias, o canhão não atira), existe uma preferência em atirar no navio que está no canhão adjacente superior. Esse método garante que todos os tiros acertem, entretanto faz com que uma boa quantidade de tiros em excesso (atiraram em navios já afundados) ocorra. Ao fim de

cada rodada, o código irá limpar todas as listas de navios e, no início do próximo turno, irá reconstruir com as informações recebidas no 'getturn'. A taxa de sucesso (navios afundados/soma de navios afundados e navios que atravessaram a última ponte) varia entre 50% e 60% em média, podendo atingir valores mais altos.

Algoritmo de retransmissão

Foi implementado uma metodologia de timeout para cada request feito no código, sendo o tempo de 0.2 segundo. Quando o timeout é atingido, a retransmissão do request ocorre, em um total de 10 vezes (cada uma com 0.2 segundos de timeout também, totalizando 2 segundos de retransmissão no máximo). Caso o máximo de retransmissão seja atingido, o cliente encerra o jogo com um erro.

Gerenciamento de comunicação entre servidores

A implementação do gerenciamento entre servidores ocorre de maneira sequencial, para todas os requests que precisam ser enviados para todos os servidores, um loop percorre o array de sockets que inclui os 4 servidores utilizados durante o jogo, e envia o mesmo request para todos. No caso do 'getturn', além de percorrer os 4 servidores, o código também repete a recepção de informação uma vez para cada ponte presente naquele jogo.

GAS utilizado

Portas: 51111, 51211, 51311

- 2020027016:0:b618cb50a9ed216eeda68cb81480545d0986339ca5374d7640f0eac2195c28ec+a041065f73268d693072bc1bb185dd8800bd103bea80814537c021ecb8b25b16