

Trabalho Prático Grau A

Algoritmos e Programação C++

OBJETIVO

Implementar software para simulação de um monitor de ambiente de rede de computadores. Este sistema deve simular as tarefas comuns de uma monitoração de rede: como manter as máquinas que se autenticaram na “rede virtual”, enviar pacotes de dados entre as estações e enviar comandos (mensagens, desligamentos, travamento e reinício) para estações. O objetivo não é utilizar comunicação em rede ou execução multi-threads, estes conceitos não serão vistos na disciplina e, portanto, não serão abordados no trabalho. Entretanto, é possível simular as funcionalidades deste ambiente de rede. Veja seção funcionamento do sistema.

Funcionamento do sistema

O programa simulador é o início da execução da simulação, ele deve iniciar um Monitor de Rede que deve gerenciar uma rede (no caso virtual). A rede receberá as conexões das estações na rede virtual. A cada evento de login, o simulador gera um usuário e solicita a autenticação ao monitor. O monitor, por sua vez, deve permitir a conexão somente de usuários válidos; e, caso o usuário seja válido solicita a inclusão de uma estação para a rede. A cada instante de simulação o Monitor fica verificando os comandos que deve executar. Quando um comando for solicitado para execução, o Monitor envia para rede, que deve fazer a sua distribuição para as devidas estações.

Quando uma estação se conecta na rede, ela recebe um id único da rede e é associada à um tempo de simulação inicial. Ela também fica no estado “ativa”. A cada tempo de simulação, o contador tempo de execução de cada estação “ativa” é incrementado em uma unidade. Quando a estação sai de execução ou é terminada pelo Monitor, um tempo de saída também é associado à estação e ela fica no estado “terminada”. Desta forma, ao final da simulação, será possível calcular o tempo médio de execução das estações, por exemplo. Também, saber qual estação entrou e saiu em que momento da rede. Além de outras informações de log para depuração.

A qualquer momento o monitor de rede pode enviar algum comando para uma determinada estação ou em broadcast. Quando o comando é direcionado para uma estação em especial, somente esta deve recebê-lo. Quando um comando é enviado broadcast, todas as estações o recebem – dependendo do comando, somente as ativas (ou apenas “bloqueadas”) podem recebê-lo. Por exemplo, se foi solicitado um comando “bloquear”, somente as estações ativas devem obedecê-lo. Já se for enviado um comando “continuar”, somente as estações no estado “bloqueada” pode executar. Os estados possíveis de uma estação são: ativa, inativa ou bloqueada.

Os comandos possíveis são: mensagem, terminar, bloquear e continuar. Pode-se associar um identificador numérico para cada tipo de comando, por exemplo: 0-mensagem, 1-terminar, 2-bloquear e 3-continuar.

Quando um comando mensagem é enviado, a estação recebe alguma informação do monitor. Quando um comando terminar é enviado, a estação recebe um tempo de fim de execução e o monitor solicita a rede que retire a estação da lista de estações conectadas. Já o comando bloquear, bloqueia a execução da estação, portanto, ela entra no estado “bloqueada”. A estação só volta a executar após o envio de um comando

“continuar” para estação.

Cada evento ocorrido na simulação deve ser impresso na tela, por exemplo: início da simulação, fim da simulação, estação A entrou na rede no tempo xxx, estação B tentou entrar na rede no tempo xxx, estação A enviou uma mensagem para C no tempo xxx. Mensagem: blablabla, ...

O objetivo final da simulação é gerar um relatório com todas as informações da simulação: tempo total de simulação, entrada e saída de cada estação, tempo de execução médio por estação, comandos enviados e situação das estações da rede a cada comando enviado.

Sobre a Geração dos Dados de Entrada

A primeira questão em aberto aqui é que, como os dados de entrada são gerados aleatoriamente, será praticamente impossível que alguma máquina consiga autenticação com login e senha aleatórios. Neste caso, duas situações são possíveis: primeira, o programador pode solicitar que o usuário digite um nome e uma senha; e, segunda, o programa pode gerar um caractere para o login e ou para a senha, inclusive associando uma possibilidade de gerar mais ou menos caracteres identificadores válidos.

Para fins de simulação, o programa deve ter um tempo máximo de simulação associado. A cada tempo de simulação, há uma probabilidade de n estações se conectarem (por exemplo, 2 à 6); em contrapartida, também há a possibilidade de n estações se desconectarem (por exemplo, 0 ou 4).

Também deve ser associada uma probabilidade de geração de comandos a cada tempo de simulação. O programador deve ainda, carregar as probabilidades de execução de um arquivo de configuração da simulação. Um layout sugerido para o arquivo é: gravar para cada tipo de evento um número mínimo e um número máximo de ocorrências por loop de simulação. Sendo assim, o programador deve associar um código numérico para cada tipo de evento. Exemplo de conteúdo para o arquivo de configuração:

0 2 7

No exemplo acima, supondo que 0 seja o identificador do evento tipo “login”, então para cada loop da simulação, devem ser gerados de 2 a 7 logins no sistema.

OBSERVAÇÕES IMPORTANTES

- NÃO É POSSÍVEL FAZER I/O (LEITURA E GRAVAÇÃO/ESCRITA) FORA DA CLASSE QUE TEM O MÉTODO MAIN;
- Será avaliada como qualidade de código a correta separação das responsabilidades de cada classe.
- Trabalho pode ser feito em dupla;
- É de responsabilidade de o aluno buscar maiores esclarecimentos e ajuda no desenvolvimento do trabalho com o professor. Normalmente, os melhores trabalhos são mostrados desde o início ao professor, que assim, pode corrigir em tempo hábil eventuais desvios de andamento;
- Só serão considerados códigos-fonte compiláveis;
- A entrega consiste em: arquivos de projeto (.h, .cpp e etc), arquivos executáveis e arquivo de modelagem;
- Distribuição dos critérios de avaliação:
 - Modelagem: 15%;
 - Funcionalidades: 60%;
 - Implementação com separação de código estilo MVC (separação da entrada/saída de

-
- dados do processamento dos dados): 15%;
 - Clareza e qualidade de código: 10%
 - O prazo de entrega é a data do GA: 03/05/16.