

Instituto Tecnológico de Aeronáutica

Departamento de Engenharia de Software
Divisão de Ciência da Computação
Laboratório de CTC-35



Proposta da atividade

Laboratório 3: Juíz Online

Alunos:

Lucas França de Oliveira, lucas.fra.oli18@gmail.com

Matheus Felipe SBF Rodrigues, matheus.felipesbf@gmail.com

Professora: Cecília de Azevedo Castro César, cecilia@ita.br

12 de setembro de 2017

1 Apresentação

Este trabalho tem como objetivo a criação de um juiz online pra ficar a serviço dos professores do ITA. Um juiz online consiste em um sistema online de correção:

1. O professor lança um laboratório e padroniza a formatação de entrada e saída do programa. Desta forma, pode-se criar um sistema automatizado de correção com casos de testes na máquina do professor ou em algum servidor do ITA.
2. O aluno submete o código fonte do um laboratório. O código fonte é compilado, executado com os casos de teste do professor e sua saída é comparada com as saídas corretas.
3. O juiz, ao final da correção, retorna um pequeno relatório gerado automaticamente pelo corretor contendo a porcentagem de casos de teste que tiveram os possíveis vereditos (Aceito, Resposta Errada, Tempo Limite Excedido, Erro em Tempo de Execução, Erro de Compilação).

O aluno não terá, assim, acesso aos casos de teste. Ao receber um veredito inesperado de Resposta Errada, por exemplo, o aluno poderá alterar o código e procurar por erros.

Isso será particularmente útil nas disciplinas de CES-10 e CES-11, em que a a entrada e saída dos laboratórios pode ser padronizada e os problemas abordados possuem solução conhecida. Essas matérias, também, são aplicadas no Curso Fundamental, abrangendo todos os alunos do ano, e possuem uma grande quantidade de laboratórios. A correção automatizada facilitará o trabalho dos professores.

2 Desgin do Juíz

2.1 Comunicação

Será usada cominuação via protocolo TCP diretamente entre o cliente e o servidor. Esse protocolo foi escolhido devido à sua simplicidade e à existência de blibliotecas na linguagem C que facilitam o trabalho. O protocolo UDP, que também possui as mesmas facilidades, é menos confiável e é usado somente quando há a necessidade de transmissão rápida de mensagens. Exemplo: em jogos online em tempo real.

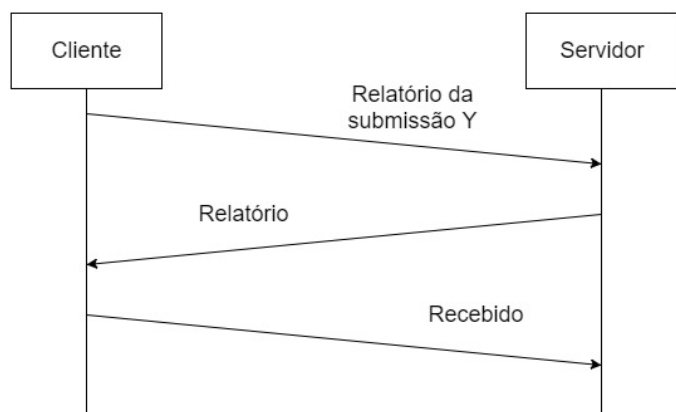
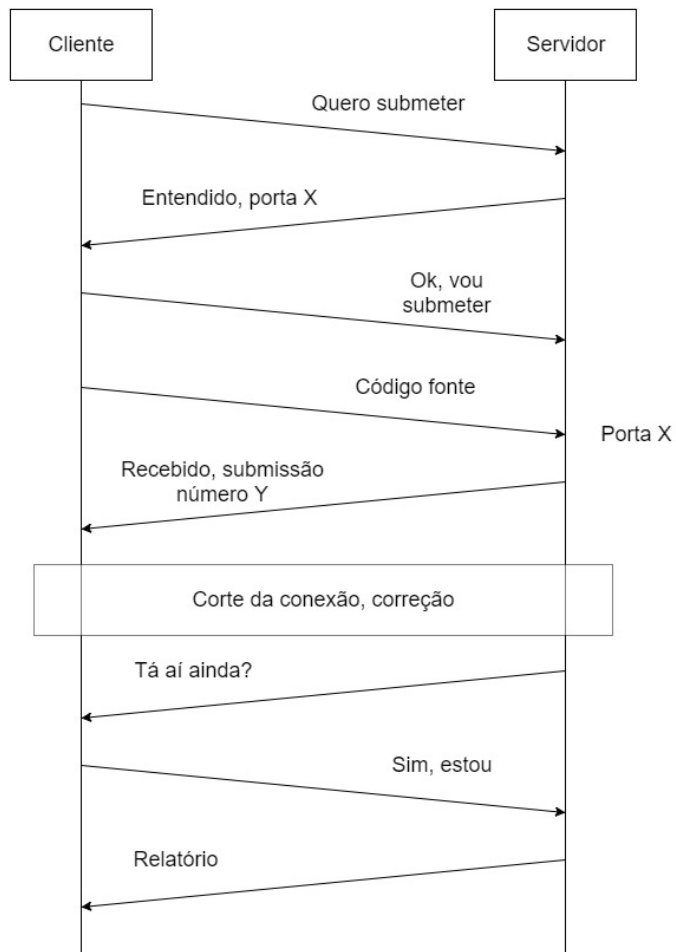
O protocolo TCP permite o envio de um buffer de bytes. O primeiro byte será um identificador de comando: cada pacote enviado segundo descrito nas seções abaixo será identificado por um identificador de comando único. O resto dos bytes será a informação daquele pacote. Exemplo: o código fonte ou a porta para o envio de dados.

Serão desenvolvidos dois comandos do lado do cliente: o submit e o report, que, respectivamente, realizam a submissão de uma solução e solicita um relatório por meio de um identificador.

O comando submit será semelhante à do protocolo FTP no modo passivo. O servidor terá um porta dedicada ao estabelecimento da conexão. O cliente se conectará com essa porta e enviará um pacote solicitando a submissão. O servidor responderá com uma nova porta para o fluxo de dados. O cliente encerra a comunicação com uma confirmação, concretizando-se o algoritmo dos “3 handshakes”.

A seguir, o cliente iniciará a conexão com o servidor na porta especificada e fará o envio do código fonte. No final, o servidor responderá que recebeu e encerrará a conexão para fazer a correção. Após a correção, o servidor tentará se conectar com o cliente e, também fazendo os “3 handshakes”, enviará o relatório.

O comando report fará apenas uma iteração dos “3 handshakes”, o que é suficiente para o recebimento do relatório.



2.2 Compilação e correção

Este laboratório tem foco na comunicação via TCP, então tentaremos simplificar o máximo possível o funcionamento da correção. Serão admitidos apenas códigos em um único arquivo em C++. A compilação ocorrerá fazendo uma chamada ao GNU g++ 6.3.

Será feita uma execução para cada teste usando o standart input e standart output, e a resposta será comparada com um gabarito. Para termos de simplicidade, o programa que será testado deve ler dois números inteiros separados por um espaço e imprimir sua soma. Os números são menores que 100000.

Não será feito o teste de tempo. Supõe-se que o laboratório do aluno não entrará em loop infinito. Erros em tempos de execução serão detectados pelo código de retorno do programa.