

Alunos: Jackson Barbosa da Silva e Samuel Lucas Vieira
Lins Barbosa

Sobre o projeto

Feito com Python, o projeto consiste de uma aplicação cliente/servidor utilizando o protocolo TCP/IP que julga algoritmos a partir de uma entrada e saída pré-estabelecidas no servidor (como os utilizados em competições de programação).

Cliente

Ao estabelecer conexão com o servidor, ao cliente são oferecidas duas opções: Ver questões e Enviar questões. Ao selecionar a primeira opção, é exibido para cada questão disponível no servidor o seu nome, um link para obter mais detalhes no site The huxley e o nome que o arquivo em C++ que estas questões deve ter. Ao selecionar a segunda opção, Enviar questões, é solicitado ao usuário que insira o nome do arquivo (Em C++) que deseja enviar (o nome do arquivo deve ser um dos que consta na lista da primeira opção.) O arquivo é então enviado ao servidor que retorna, para cada caso de teste, se o usuário acertou ou errou aquele caso de teste.

Servidor

Ao iniciar o servidor fica escutando uma porta específica definida ao rodar o servidor, ao conectar-se o servidor adiciona os dados do cliente em uma lista com todos os clientes conectados. Os clientes conectados podem solicitar ver as questões cadastradas, ao fazer isso, o servidor envia ao cliente a lista de questões cadastradas no servidor, para adicionar uma questão deve-se adicionar a entrada válida e a saída esperada para ela na pasta do servidor, seguindo o padrão: [nome_do_arquivo_esperado]_inp e [nome_do_arquivo_esperado]_judge_out, após fazer isso concatena-se mais uma linha a string com a lista das questões especificando o link para a questão, o nome dela e o nome do arquivo que deve ser enviado. Ao ser solicitada julgar a questão, o servidor usa o código do judge pré-escrito para compilar o arquivo, depois para executar o arquivo e salvar a saída obtida no arquivo [nome_do_arquivo]_user_out, compara-se a saída cadastrada com a gerada pelo programa do usuário, apaga os arquivos temporários, que são: o arquivo c++ enviado pelo cliente, a saída gerada pelo cliente e o executável gerado ao compilar o arquivo enviado, por fim, envia ao cliente o julgamento da questão para o caso de teste.

Dificuldades

Ao criar o servidor, inicialmente criamos conexões em portas diferentes para aceitar mais de um cliente (cada um se conectando a uma porta), depois conseguimos implementar de forma que cada cliente se conecta a mesma porta do servidor.

Possíveis melhorias

Com o propósito de melhorar a experiência de usuário, algumas alterações e adições poderiam ser feitas: A primeira delas seria uma interface de usuário, tornando assim a aplicação mais amigável para o uso. Outra alteração seria um sistema que automatizasse e facilitasse o cadastro de questões, visto que na implementação atual esse processo é feito manualmente. Além disso, também poderia ser aumentada a quantidade de linguagens suportadas pela aplicação, que atualmente só suporta C ++, incluindo linguagens como Python e Java. Por último, outra alteração interessante seria a forma como o servidor analisa os casos de teste, ao invés de apenas apontar se obteve a resposta correta ou não, seria bom apontar qual foi o erro que ocorreu, incluindo casos como Tempo Limite Excedido (TLE), Erro de Apresentação, Erro de Compilação, dentre outros.