Universidade Federal do Ceará
Campus de Russas

Disciplina Redes de Computadores

Trabalho prático - Sockets

Informações Gerais

Neste trabalho, o grupo de alunos (03 alunos no máximo) deverá desenvolver e implementar um protocolo de camada de aplicação para a aplicação proposta: Estufa inteligente (especificação encontrada no final deste documento).

O grupo deverá seguir os requisitos apresentados para a aplicação escolhida. O *core* do protocolo consiste nas mensagens trocadas para o funcionamento de cada aplicação. O trabalho possui duas etapas: desenvolvimento do protocolo e implementação prática.

1. Etapa de desenvolvimento

No desenvolvimento, o grupo deverá criar mensagens que serão trocadas na camada de aplicação pelos componentes do sistema escolhido de forma que os requisitos de funcionamento sejam atendidos.

Cada mensagem deverá ser documentada separadamente, explicitando a sua estrutura (mensagens com a mesma função como por exemplo, mensagens de dados de sensores podem ser documentadas juntas). Por exemplo, um protocolo para troca de e-mails entre dois usuários pode conter uma mensagem chamada ENVIAR_EMAIL, com os campos:

1. e-mail do remetente (obrigatório)
2. número de destinatários (obrigatório)
3. e-mail(s) do(s) destinatário(s) 1...n (obrigatório)
4. corpo da mensagem (opcional)
5. prioridade (opcional)

A forma como a troca de mensagens entre os componentes atende aos requisitos de funcionamento do sistema poderá ser descrita por meio de texto ou imagens. É válido frisar que nem todo requisito de funcionamento será atendido por uma mensagem; é possível que o requisito seja atendido por um campo de uma mensagem específica.

O protocolo deverá possuir um *header* específico para identificá-lo. O grupo pode consultar *headers* de outros protocolos da camada de aplicação (HTTP, SMTP etc.) para inspiração.

2. Etapa de implementação

Nesta etapa, o grupo deverá implementar o protocolo desenvolvido na etapa anterior utilizando sockets. Nesta etapa, as seguintes regras se aplicam:

- Não há restrições quanto à linguagem de programação utilizada; várias delas (como C/C++, Java e Python) têm interface de sockets já implementadas em bibliotecas.
- A implementação de cada processo é livre, podendo ter mais de um thread.
- O grupo deverá explicitar o SO e o compilador utilizados para que o trabalho possa ser devidamente avaliado.
- O software deverá explicitar a troca de mensagens (imprimindo na tela, por exemplo) e implementar as situações que mostram os diversos requisitos de funcionamento.

Avaliação

O trabalho dos grupos será avaliado segundo os seguintes critérios:

- Cumprimento dos requisitos da aplicação (3,0 pts)
- Descrição do protocolo desenvolvido (2,0 pts)
- Funcionamento da aplicação seguindo as regras estabelecidas (3,0 pts)
- Clareza da implementação (documentação do código) (2,0 pts)

Entrega: 09/05/2024 -- 23h59

Cada grupo deverá enviar pelo SIGAA um relatório contendo:

- Integrantes do grupo
- Aplicação escolhida
- Descrição do protocolo desenvolvido, com descrição do *header*, mensagens e como os requisitos funcionais são cumpridos.
- o código fonte da implementação do protocolo em sockets, juntamente com as informações de SO/compilador utilizados para avaliação
- Vídeo de até 8 minutos explicando o funcionamento da aplicação.

Observações: O relatório não precisa ser rebuscado! Sejam claros e assertivos na descrição do protocolo.

Façam um código bem documentado e usem nomes bem descritivos nas variáveis. Grupos que entregarem códigos mal documentados ou de difícil compreensão perderem pontos no critério Clareza da implementação.

Aplicação: Estufa Inteligente

Estufas são utilizadas no cultivo de plantas em condições controladas de temperatura, umidade, luminosidade, nível de CO_2, entre outras variáveis. Nesta aplicação, as condições da estufa são configuradas, monitoradas e controladas por um gerenciador, que se comunica com os sensores/atuadores dentro da estufa e pode receber configurações ou responder consultas de um cliente externo.

Componentes

- *Sensores*:

– Temperatura interna

– Umidade do solo

– Nível de CO_2

- *Atuadores*:

– Aquecedor (quando ligado, aumenta a temperatura)

– Resfriador (quando ligado, diminui a temperatura)

– Sistema de irrigação (quando ligado, aumenta a umidade do solo)

– Injetor de CO_2 (quando ligado, aumenta a concentração do gás)

- *Gerenciador*, que funciona como o servidor da aplicação.
- *Cliente*, que pode configurar os parâmetros da estufa e requisitar valores das leituras dos sensores.

Princípio de operação

O Gerenciador deve manter as leituras dos sensores entre os valores máximo e mínimo configurados.

Requisitos Funcionais

1. Sensoriamento:

    1.1. Todos os sensores têm um identificador único.

    1.2. Os sensores devem se conectar ao Gerenciador e se identificar.

    1.3. Após receber confirmação, os sensores deverão enviar sua leitura a cada 1s ao Gerenciador.

2. Atuadores:

    2.1. Todos os atuadores têm um identificador único.

    2.2. Os atuadores devem se conectar ao Gerenciador e se identificar.

    2.3. Os atuadores poderão ser ligados ou desligados pelo Gerenciador.

3. Gerenciador:

3.1. O Gerenciador deverá aceitar a conexão de sensores e atuadores do sistema.

3.2. O Gerenciador deve receber as leituras de todos os sensores do sistema e armazenar o último valor recebido.

3.3. O Gerenciador deve ligar ou desligar os atuadores caso os valores das leituras de sensores indicarem que as variáveis estão fora dos valores máximo e mínimo configurados.[1]

3.4. O Gerenciador deve ser capaz de fornecer ao Cliente a última leitura de cada sensor do sistema quando receber uma requisição.

4. Monitoramento: o Cliente deve ser capaz de requisitar a última leitura de qualquer sensor do sistema ao Gerenciador.

Agradecimentos:

Prof. Kalinka Regina Lucas Jaquie Castelo Branco por disponibilizar o trabalho.

[1] Para diminuir a quantidade de atuações nos limites máximo e mínimo, é possível fazer um *controle por histerese*