

Paradigmas de Sistemas Distribuídos

Teste¹

30 de maio de 2022

Duração: 1h15m

I

1 Justifique a utilidade do paradigma de programação por eventos, nas suas várias formas, do ponto de vista da eficiência para um servidor em grande escala.

2 Considere um sistema de automação para domótica em que é possível estabelecer rotinas tais como: consultar um termómetro remoto; registar o valor obtido num servidor de bases de dados; dependendo do valor lido, contactar o controlador da caldeira para aquecer; verificar de novo o termómetro com intervalos de 1 minuto até atingir o valor desejado; desligar então a caldeira; e registar a terminação no servidor de bases de dados. Ordene de acordo com a conveniência para o programador na resolução deste problema, justificando sucintamente, os mecanismos de programação por eventos que estudou.

II

1 É possível ter único processo Erlang a receber dados vindos de vários sockets TCP. Descreva as vantagens de usar a abordagem alternativa, de dedicar um processo Erlang a cada conexão.

2 Considere o seguinte fragmento do código de um processo Erlang, que pode receber dois “tipos” de mensagens e pretende o seu processamento.

```
loop(State) ->
  receive
    {foo, Data} -> NewState = foo(Data), loop(NewState);
    {bar, Data} -> NewState = bar(Data), loop(NewState)
  end.
```

Explique o que entende por dar prioridade a um “tipo” de mensagem, e em que circunstâncias tal é relevante. Diga, justificando, se o código acima dá prioridade a mensagens do tipo `foo` e, se não for esse o caso, apresente código apropriado a tal fim.

3 Descreva em que situações utilizaria, em ZeroMQ, o padrão PUSH-PULL e o padrão REQ-REP. Explique as limitações destes 4 tipos de sockets ZeroMQ, para um cenário com troca de mensagens, entre vários “clientes” e um “servidor”, algumas apenas para processamento e outras que exigem resposta. Descreva que tipos de sockets ZeroMQ, e como os utilizaria, neste cenário, para poder ter uma única thread “servidora”, sem utilizar *poll*.