DCC023 – Redes de computadores

Trabalho prático 0 –

Professores: Daniel Fernandes Macedo, Marcos Augusto M. Vieira

Monitor: Lucas Augusto Maia da Silva

Data de entrega: 05/10/2015

Trabalho em grupos de dois alunos (ou individual).

Valor do trabalho: 5 pontos

1. Introdução

No Filme "Os Estagiários", dois vendedores decidem aplicar para um estágio no Google. Durante o processo de seleção, uma das tarefas é escrever um aplicativo popular para celular. Os dois propõem um aplicativo que "barra" o envio de mensagens por pessoas bêbadas ao pedir que a pessoa resolva um problema matemático.

Neste trabalho vocês vão fazer um trabalho similar, mas em rede, para evitar que os alunos enviem trabalhos práticos quando estiverem cansados, e assim cometerem erros (por exemplo, enviar o zip com o código errado, etc). Um servidor vai propor um desafio matemático, e o cliente deve responder esse desafio. Caso o cliente responda corretamente, o servidor irá enviar uma mensagem de sucesso ao cliente.

2. Programas a serem desenvolvidos

O trabalho prático consistirá na implementação de dois programas, o cliente e o servidor. Ambos vão receber dois parâmetros de entrada, como mostrado a seguir:

```
cliente <ip/nome> <porta>

servidor <porta>
```

O primeiro programa, o cliente, irá conectar no servidor definido pelo endereço IP (ou nome, por exemplo *login.dcc.ufmg.br*) e porta passados por parâmetro.

Já o servidor irá executar um serviço, que necessariamente irá tratar várias conexões e só sair depois que o usuário apertar control-c (o servidor não precisa suportar conexões simultâneas, mas ele deve suportar que um cliente conecte, e em seguida outro, assim sucessivamente, sem que ele tenha que ser executado novamente). A porta a ser escutada é dada pelo parâmetro único do programa.

Segue abaixo um exemplo de execução do programa. A mensagem de texto a ser impressa pode ser diferente na sua versão, entretanto o cliente deve apresentar um desafio (a conta a ser resolvida) em forma textual, bem como uma mensagem de sucesso ou erro do desafio:

```
damacedo@pc-> ./cliente servidor.winet.dcc.ufmg.br 1234
Por favor responda quanto é 4 * 5:
20
Sucesso! Autenticacao liberada.
damacedo@pc->./cliente servidor.winet.dcc.ufmg.br 1234
Por favor responda quanto é 20 ** 2:
4001
Erro! Voce esta muito cansado, tente novamente mais tarde.
Por favor responda quanto é 240 + 7:
...
```

Reparem que o servidor irá continuar enviando desafios enquanto o cliente não acertar a resposta, ou o cliente fechar a conexão. Os desafios devem mudar a cada tentativa. A resposta do cliente é enviada ao servidor, que computa o desafio e informa se o cliente acertou.

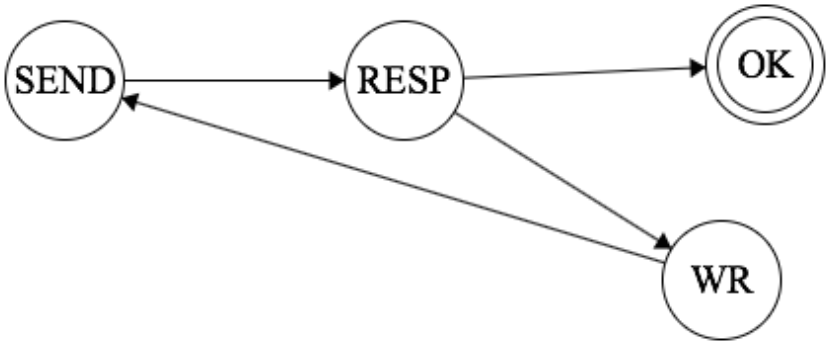
As operações matemáticas usadas no desafio serão subtração ("-"), adição ("+"), multiplicação ("*") e exponenciação ("**"). Todas envolverão dois números. Desafios com mais de uma operação matemática não serão suportados. Somente números inteiros não negativos serão aceitos. Caso alguma operação gere um valor negativo, o resultado deve ser enviado como ZERO.

Associado ao programa, existe o protocolo Desafio 2.0 (houve um desafio 1.0 proposto pelos professores, mas devido a uma parte ambígua na especificação, vocês vão implementar a versão 2.0), que é descrito abaixo. Ele possui 4 mensagens, a saber:

- SEND: envia um desafio do servidor para o cliente.

- RESP: O cliente envia a resposta para o servidor.
- OK: O servidor indica que a resposta foi correta.
- WR: O servidor indica que a resposta foi incorreta.

A figura abaixo mostra uma máquina de estados do protocolo, que começa no estado SEND.



As mensagens possuem o seguinte formato:

SEND:

Código (1 byte) 0x01	OPERANDO1 (2 bytes)	OPERANDO2 (2 bytes)	OPERACAO (1 byte)

As operações deverão ser mapeadas para os seguintes códigos:

- 0x0 – subtração
- 0x1 – adição
- 0x2 – multiplicação
- 0x3 – exponenciação

RESP: Esta mensagem terá a resposta, que deverá ser zero se o desafio gerar um valor negativo, e possui os seguintes campos:

Código (1 byte) 0x02	RESPOSTA (2 bytes)

As mensagens OK e WR terão somente um byte, o código da operação, que será 0x03 e 0x04, respectivamente.

3. Entrega do código

O código e a documentação devem ser entregues em um arquivo Zip (não pode ser RAR nem .tgz nem .tar.gz) no Moodle, contendo um arquivo **readme.txt** com o nome dos integrantes, e um arquivo PDF da documentação. Incluam todos os

arquivos (.c, .h, makefile, não incluam executáveis ou arquivos objeto) EM UM ÚNICO DIRETÓRIO. Um makefile deve ser fornecido para a compilação do código.

Parte desse trabalho envolve o aprendizado de como construir um makefile e utilizar a ferramenta Make. Este makefile, quando executado sem parâmetros, irá gerar os dois programas, *cliente* e *servidor*, EXATAMENTE com esses nomes. Submissões onde os programas não seguem as especificações de parâmetros e nomes, makefiles não funcionando ou arquivos necessários faltando não serão corrigidos. Programas que não compilarem também não serão corrigidos. O julgamento do trabalho será feito em um computador rodando o SO Linux.

4. Documentação

A documentação, além de ser entregue com o Zip junto ao código, deve ser entregue impressa ao professor até o dia seguinte à entrega, na sala de aula. Caso não tenhamos aula no dia seguinte ao prazo para envio, iremos combinar antecipadamente qual é a forma mais conveniente de entrega da documentação. Trabalhos que forem entregues no Moodle mas sem a documentação impressa **não serão** corrigidos.

O texto da documentação deve ser breve, de forma que o corretor possa entender o que foi feito no código sem ter que entender linha a linha dos arquivos. Implementações modularizadas deverão mencionar quais funções são implementadas em cada módulo ou classe. A documentação deve conter os seguintes itens:

- Sumário do problema a ser tratado
- Uma descrição sucinta dos algoritmos e dos tipos abstratos de dados, das principais funções, e procedimentos e as decisões de implementação
- Decisões de implementação que porventura estejam omissos na especificação
- Testes, mostrando que o programa está funcionando de acordo com a especificação, seguidos da sua análise
- Conclusão e referências bibliográficas

5. Avaliação

A avaliação do trabalho será composta pela execução dos programas desenvolvidos e pela análise da documentação. Os seguintes itens serão avaliados:

- A qualidade do código (código bem organizado, com comentários explicativos, variáveis com nomes intuitivos, modularidade, etc).
- Execução correta do código em entradas de testes, a serem definidas no momento da avaliação. As entradas de teste irão exercitar a funcionalidade completa do código e testar casos especiais ou de maior dificuldade de implementação, mas que devem ser tratados por um programa correto.

- Aderência ao protocolo especificado. Iremos usar ferramentas de captura do tráfego da rede (*tcpdump*, *wireshark*) e iremos analisar o código para verificar se a comunicação está implementada da forma definida na documentação.

- Conteúdo da documentação, que deve conter os itens mencionados anteriormente.

- Coerência e coesão da documentação (apresentação visual e organização, uso correto da língua, qualidade textual e facilidade de compreensão).

Como mencionado anteriormente, trabalhos fora da especificação (por exemplo, sem makefile, que não gerem programas com os nomes especificados, que não compilem ou não possuam os parâmetros esperados) não serão corrigidos. Trabalhos fora da especificação tomam muito trabalho do corretor, que deve entender como compilar e rodar **cada programa avaliado**, tempo esse que deveria ser empregado para avaliar a execução e corretude do código.

Qualquer elemento que não esteja especificado neste documento, mas que tenha que ser inserido para que o protocolo funcione, deve ser descrito na documentação de forma explícita.

6. Desconto de nota por atraso

Os trabalhos poderão ser entregues até a meia-noite do dia especificado para a entrega. O horário de entrega deve respeitar o relógio do sistema Moodle, ou seja, a partir de 00:01 do dia seguinte à entrega no relógio do Moodle, os trabalhos já estarão sujeitos a penalidades.

A fórmula para desconto por atraso na entrega do trabalho prático ou resenha é:

Desconto = $2^{d-1}/0.32$ %

onde *d* é o atraso em dias úteis. Note que após 5 dias úteis, o trabalho não pode ser mais entregue.

7. Referências

Programação em C:

- http://www.dcc.ufla.br/~giacomin/Textos/tutorialc.pdf
- ftp://ftp.unicamp.br/pub/apoio/treinamentos/linguagens/c.pdf
- http://www.cs.cf.ac.uk/Dave/C/CE.html
- http://www2.its.strath.ac.uk/courses/c/
- http://www.lysator.liu.se/c/bwk-tutor.html

Uso do programa make e escrita de Makefiles:

- http://comp.ist.utl.pt/ec-aed/PDFs/make.pdf

- http://haxent.com.br/people/ruda/make.html
- http://informatica.hsw.uol.com.br/programacao-em-c16.htm
- http://www.gnu.org/software/make/manual/make.html
- http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/