



Fundamentos de Redes de Computadores

Trabalho 01

Prof. Tiago Alves

Camada de Aplicação: Cliente DNS

Introdução

A disciplina de Fundamentos de Redes de Computadores trata de diversos tópicos relativos a Sistemas Operacionais, sistemas que provêm uma forma intuitiva de se utilizar as funcionalidades de computadores digitais sem que seja necessário ao usuário ou programador ter profundo conhecimento das interações entre os diferentes *hardwares* que compõem um computador.

Para construir ou adicionar funcionalidades a esses sistemas computacionais, é necessário conhecimento de linguagens de programação e ferramentas de desenvolvimento.

Objetivos

- 1) Exercitar conceitos de programação de sistemas operacionais.
- 2) Estudar a implementação de esquemas de comunicações entre processos (IPC), especificamente a utilização da interface *socket*.

Referências Teóricas

Mitchell, Mark, Jeffrey Oldham, and Alex Samuel. *Advanced linux programming*. New Riders, 2001.

Material Necessário

- Computador com sistema operacional programável
- Ferramentas de desenvolvimento GNU/Linux ou similares: compilador GCC, depurador, editor de texto.

Roteiro

- 1) Revisão de técnicas e ferramentas de desenvolvimento de aplicações para o sistema operacional Linux.

Colete o material acompanhante do roteiro do trabalho a partir do Moodle da disciplina e estude os princípios e técnicas de desenvolvimento de aplicações para o sistema operacional Linux.

- 2) Realizar as implementações solicitadas no questionário do trabalho.



Implementações e Questões para Estudo

- 1) Escreva um cliente para o protocolo **DNS**.
 - O **cliente** receberá entradas do teclado e tentará realizar o envio de uma requisição DNS (*RFC1034 e RFC1035*).
 - O primeiro argumento da linha de comando deverá informar o endereço o nome cuja resolução se procura. O segundo argumento informará o IP do servidor DNS que será consultado.
 - O servidor funcionará na porta padrão para o serviço DNS: UDP 53
 - *Firewalls* podem impedir com que pacotes com as consultas alcancem servidores localizados fora da rede de acesso.
 - Em relação ao formato do payload UDP da consulta, seguem orientações:
 - O **cliente** deverá realizar apenas consultas do tipo MX (mail exchanger record).
 - Transaction ID (16 bits): o cliente deverá gerar um **número aleatório de 16 bits** para identificar a transação/consulta.
 - Flags (16 bits): as flags de consulta deverão indicar uma consulta recursiva. Dessa forma, os 16 bits de deverão estar codificados em **0x0100**.
 - Question (16 bits): deverá indicar apenas 1 consulta e, desta forma, deverá estar codificado em **0x0001**.
 - Answer RRs, Authority RRs, Additional RRs (todos os três são campos de 16 bits): deverão estar codificados em **0x0000**.
 - Query: a query deverá informar uma consulta Type MX, Class IN para o Name informado como primeiro parâmetro da linha de comando.
 - A implementação do cliente deverá utilizar o protocolo UDP como protocolo de transporte.
 - Como o protocolo UDP é orientado a mensagem, o **cliente** deverá aguardar 2 segundos pela chegada da resposta. Caso a resposta não tenha chegado, uma nova tentativa deverá ser realizada.
 - O **cliente** tentará resolver o nome por até 3 vezes. Depois da terceira vez, uma mensagem de erro deverá ser informada ao usuário.
 - Caso o nome apresentado pelo usuário não exista,
 - O resultado da consulta DNS deverá ser informado na saída padrão (stdout do console) e deverá ter o seguinte formato:
 - nome_dominio <> nome_servidor_email
 - Para informar o resultado da consulta, será necessário interpretar a mensagem retornada pelo servidor e extrair a informação do nome do servidor de e-mail de seu payload.
 - Exemplos de interação:

```
# Resolução bem sucedida
$ ./meu_cliente unb.br 8.8.8.8
unb.br <> unb-br.mail.protection.outlook.com
```

```
# Resolução com falha: nome de domínio não existe
$ ./meu_cliente imagdaskdasdasj.br 1.1.1.1
Domínio imagdaskdasdasj.br nao encontrado
```

```
# Resolução com falha: domínio não possui entrada MX
$ ./meu_cliente fga.unb.br 8.8.8.8
Domínio fga.unb.br nao possui entrada MX
```

```
# Resolução com falha: servidor não existe/não atendeu
$ ./meu_cliente unb.br 1.2.3.4
Nao foi possível coletar entrada MX para unb.br
```



- Restrições:
 - Para este trabalho, será admitida a utilização de bibliotecas auxiliares para para suporte à implementação. Quando se tratar de serviços de comunicação, apenas bibliotecas de socket UDP da sua linguagem preferida de programação poderão ser utilizadas.
 - Bibliotecas auxiliares serão permitidas



Instruções e Recomendações

A submissão das respostas aos problemas dos trabalhos deverá ser feita através do Moodle da disciplina.

Cada resposta a problema desse Trabalho **01** deverá ser entregue em um pacote ZIP. A dupla de alunos deverá nomear o pacote ZIP da seguinte forma: nome_sobrenome_matricula_nome_sobrenome_matricula_**trab01**.zip.

Entre os artefatos esperados, listam-se:

- códigos-fonte das soluções dos problemas, correspondendo a **80% dos escores de pontuação final** do trabalho;
- documentação mínima da aplicação, correspondendo a **20% dos escores de pontuação final** do trabalho:
 - o qual sistema operacional foi usado na construção do sistema;
 - o qual ambiente de desenvolvimento foi usado;
 - o como construir a aplicação;
 - o como executar a aplicação;
 - o quais são as telas (instruções de uso);
 - o quais são as limitações conhecidas

Não devem ser submetidos executáveis.

Códigos-fonte com erros de compilação/interpretação serão desconsiderados (anulados).

Os trabalhos poderão ser realizados em duplas; a identificação de cópia ou plágio irá provocar anulação de todos os artefatos em recorrência.

Em relação ao uso das linguagens de programação, as equipes deverão atentar-se para a seguinte tabela de deflatores de escore:

- linguagem compilada C: **100%** de escores totais do trabalho;
- linguagem compilada C++: **95%** de escores totais do trabalho;
- linguagens interpretadas e Java: **90%** de escores totais do trabalho.

ATENÇÃO: Não será permitido o uso de bibliotecas de comunicações para a solução do trabalho. Ou seja, o payload do pacote com requisição DNS deverá ser montado programaticamente. O pacote de resposta poderá usar funções para manipulação dos campos e extração da resposta à consulta.