



Fundamentos de Redes de Computadores

Trabalho 01

Prof. Tiago Alves

Camada de Aplicação: Cliente de Tempo (SNTP)

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 **SNTP**.

- O **cliente** receberá entradas do teclado e tentará realizar o envio de uma requisição **SNTP** (*Simple Network Time Protocol, RFC1769*).
 - O primeiro argumento da linha de comando deverá informar o endereço IP do servidor **SNTP** que será consultado.
 - Uma mensagem **NTP** consiste de uma sequência de 384 bits ou uma estrutura de dados de 48 bytes contendo 17 campos.

```
typedef struct
{
    uint8_t li_vn_mode;           // Eight bits. li, vn, and mode.
                                   // li.   Two bits.   Leap indicator.
                                   // vn.   Three bits. Version number of the protocol.
                                   // mode. Three bits. Client will pick mode 3 for client.

    uint8_t stratum;              // Eight bits. Stratum level of the local clock.
    uint8_t poll;                 // Eight bits. Maximum interval between successive messages.
    uint8_t precision;            // Eight bits. Precision of the local clock.

    uint32_t rootDelay;           // 32 bits. Total round trip delay time.
    uint32_t rootDispersion;      // 32 bits. Max error aloud from primary clock source.
    uint32_t refId;               // 32 bits. Reference clock identifier.

    uint32_t refTm_s;             // 32 bits. Reference time-stamp seconds.
    uint32_t refTm_f;             // 32 bits. Reference time-stamp fraction of a second.

    uint32_t origTm_s;            // 32 bits. Originate time-stamp seconds.
    uint32_t origTm_f;            // 32 bits. Originate time-stamp fraction of a second.

    uint32_t rxTm_s;              // 32 bits. Received time-stamp seconds.
    uint32_t rxTm_f;              // 32 bits. Received time-stamp fraction of a second.

    uint32_t txTm_s;              // 32 bits and the most important field the client cares
    about. Transmit time-stamp seconds.
    uint32_t txTm_f;              // 32 bits. Transmit time-stamp fraction of a second.
} ntp_packet;                    // Total: 384 bits or 48 bytes.
```

- Para uma requisição simples, você deverá zerar a string de 48 bytes e setar o primeiro byte da string 0x1B, o que significa carregar os seguintes valores: **li = 0**, **vn = 3** e **mode = 3**
- Em seguida, um **socket** **UDP** deverá ser criado para encaminhar a mensagem para a porta 123 do servidor **NTP**, cujo IP foi informado em parâmetro da linha de comando.
- Use o payload de 48 bytes para solicitar informação de tempo ao servidor através do **socket** **UDP** e aguarde pela resposta do servidor.
- Uma vez recebida a mensagem do servidor, realize o *parsing* (interpretação) da informação de tempo encaminhada pelo servidor. Realize as devidas conversões de forma a apresentar, em saída em console, uma informação no seguinte formato:
Data/hora: Qui Mar 28 23:11:16 2019
- Caso o cliente não obtenha resposta em 20 segundos, ele deverá tentar novamente e aguardar mais 20 segundos. Se após a segunda tentativa a resposta não for recebida, o cliente deverá informar a seguinte mensagem em console:
Data/hora: não foi possível contactar servidor



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 limites de escore:

- linguagem compilada C: **100%** de escores da parcela de implementação;
- linguagem compilada C++: **95%** de escores da parcela de implementação;
- linguagem Java: **90%** de escores da parcela de implementação;
- linguagens interpretadas: **90%** de escores da parcela de implementação.

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