# Trabalho 4 – Análise do conteúdo de uma página web

A comunicação entre computadores é feita via endereços, assim como em um sistema de correios. Assim como cada casa tem um endereço (Rua João, n.o 3), cada computador também tem um endereço denominado endereço IP (endereço de Protocolo Internet). Por exemplo, considere dois computadores em rede:

Computador 1: Endereço 192.168.0.2
Computador 2: Endereço 192.168.0.3

Assim, quando o Computador 1 deseja comunicar com o Computador 2, basta ele avisar que deseja abrir uma conexão e trocar mensagens com tal máquina.

Da mesma forma ocorre na Internet. Por exemplo, podemos descobrir o endereço IP de um computador na Internet da seguinte maneira:

ping www.icmc.usp.br

Repare que irá aparecer um endereço na tela de seu computador. Esse é o endereço do computador que responde pelas páginas WEB do ICMC/USP.

Agora considere sites de notícia tais como o Slashdot.org. Esse site tem um página com o resumo de todas notícias para algumas categorias, tais como:

http://rss.slashdot.org/Slashdot/slashdotGames
http://rss.slashdot.org/Slashdot/slashdotAskSlashdot
http://rss.slashdot.org/Slashdot/slashdotYourRightsOnline
http://rss.slashdot.org/Slashdot/slashdotPolitics
http://rss.slashdot.org/Slashdot/slashdotLinux
http://rss.slashdot.org/Slashdot/slashdotDevelopers

Perceba que cada uma das páginas acima contém uma série de notícias. Acesse qualquer uma delas. Caso o código fonte não apareça, clique sobre a página com o botão direito de seu mouse. Clique depois em “View Page Source” ou “Ver Código da Página”. Você verá as instruções em um formato chamado XML. Esse formato é comum para montar conteúdo e disponibilizar na Internet.

Olhando ainda o código da página, observe que haverá *tags* ou comandos tais como:

```
<title>
</title>
<description>
</description>
```

Perceba que há conteúdo entre essas tags. Por exemplo:

```
<title>Título de uma notícia</title>
<description>Descrição ou corpo da notícia</description>
```

Perceba que cada página contém um longo conjunto de notícias e de seus respectivos corpos ou descrições. O objetivo deste trabalho é analisar o conteúdo de uma página de notícias, gerando uma espécie de resumo das informações contidas nessa página, utilizando para isso tags encontradas.

O programa deverá, dado o endereço IP de um servidor (por exemplo, do Slashdot é 74.125.137.121), a porta de acesso, e um caminho para uma página (por exemplo, http://rss.slashdot.org/Slashdot/slashdotGames):

1) Solicitar a página, recebendo a cadeia de caracteres (string) referente à página solicitada
2) Identificar todos os títulos de notícias (os quais estarão contidos em tags <title></title>), armazenando-os numa estrutura de dados.
    - Não deverão ser armazenados títulos duplicados.
    - Considere que entre as tags <title> </title> não haverá outras tags
3) Ordenar esses títulos (strings) de maneira crescente
4) Contar o número de palavras existentes no corpo ou descrição da notícia (contidos nas tags <description></description> que estiverem logo após o título). Caso a notícia não tiver descrição, deverá ser considerado que a descrição da notícia possui 0 (zero) palavras. Caso um mesmo título apareça mais que uma vez, o número de palavras deve ser acumulado.
    - Deverá ser contado como "palavra" qualquer sequência (cadeia) de caracteres delimitadas pelos seguintes caracteres:
        1. espaço em branco ( )
        2. caracter vírgula (,)
        3. caracter menor (<)
        4. caracter maior (>)

As quatro operações acima não precisam ser feitas nessa ordem exata, podendo ser feitas em conjunto quando possível.

Apresente na tela do computador:

1) O número total de notícias encontradas (por meio dos títulos)
2) O número total de notícias encontradas (por meio dos títulos) cujo número de palavras na descrição seja maior que zero
3) A lista de títulos ordenada
4) O número de palavras encontradas na descrição de cada notícia

Para obter a string referente à página solicidata, o código abaixo faz conexão com qualquer servidor e permite que obtenhamos como resposta o conteúdo da página.

Sendo assim, o trabalho ficará apenas na análise do conteúdo da página, o qual é representado por uma cadeia de caracteres.

Para executar esse código nas páginas do Slashdot utilize, por exemplo:

./client 74.125.137.121 80 http://rss.slashdot.org/Slashdot/slashdotGames

Salvar o conteúdo abaixo como client.c e compilar da seguinte maneira:

gcc -o client client.c

```
/* Nome do arquivo client.c */

#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <arpa/inet.h>

char *getpage(char *ip, int port, char *page) {

    int sockfd = 0, n = 0, count = 0;
    char *recvBuff = NULL, *sendBuff;
    struct sockaddr_in serv_addr;

    if((sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
    {
        printf("\n Error : Could not create socket \n");
        return NULL;
    }

    memset(&serv_addr, '0', sizeof(serv_addr));

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(port);

    if(inet_pton(AF_INET, ip, &serv_addr.sin_addr)<=0)
    {
        printf("\n inet_pton error occured\n");
        return NULL;
    }

    if( connect(sockfd, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) < 0)
    {
       printf("\n Error : Connect Failed \n");
       return NULL;
    }

    // requesting the page
    // building the request string
    sendBuff = (char *) malloc(sizeof(char) * (strlen(page) + 7));
    snprintf(sendBuff, sizeof(char) * (strlen(page) + 7), "GET %s\r\n", page);

    // writing in the socket the request
    write(sockfd, sendBuff, strlen(sendBuff));
```

```
    // reading the page
    int i = 0;

    count = 1;
    recvBuff = (char *) realloc(recvBuff, sizeof(char));

    // read each character of the page
    while ( (n = read(sockfd, &recvBuff[count-1], 1)) > 0)
    {
       //printf("%c", recvBuff[count]);
       recvBuff = (char *) realloc(recvBuff, sizeof(char) * (count+1));
       count++;
    }

    recvBuff[count-1] = '\0';

    free(sendBuff);

    return (recvBuff);
}



int main(int argc, char *argv[])
{
    char *ip, *page, *data;
    int port;

    if(argc != 4)
    {
        printf("\n Usage: %s <ip of server> <port> <page>\n",argv[0]);
        return 1;
    }

    ip = argv[1];
    port = atoi(argv[2]);
    page = argv[3];

    data = getpage(ip, port, page);

    printf("%s\n", data);

    free(data);

    return 0;
}
```

## Exemplo de entrada e saída

No trabalho as entradas serão fornecidas pelo usuário após a execução do programa. A primeira linha corresponde ao IP, a segunda à porta de acesso, e a terceira ao caminho da página. Respeitando o formato:

`IP\n`

`PORTA\n`

`ENDERECO\n`

Exemplo de entrada:

`74.125.234.191\n`

`80\n`

`http://news.google.com.br/news/feeds?output=rss\n`

em que `\n` significa uma quebra de linha.

As saídas produzidas pelo programa devem respeitar o formato:

```
numero-N-de-noticias-encontradas\n
numero-N-de-noticias-encontradas-com-uma-ou-mais-palavras-na-descricao\n
titulo-noticia1_:_numpalavras-descricao-noticia1\n
titulo-noticia2_:_numpalavras-descricao-noticia2\n
…
titulo-noticiaN_:_numpalavras-descricao-noticiaN\n
```

Exemplo de saída:

```
11\n
10\n
Avião faz pouso forçado após decolar de São Paulo - Paraná-Online_:_153\n
BC decreta liquidação extrajudicial dos bancos Cruzeiro do Sul e ... - Globo.com_:_155\n
Bento XVI condena no Líbano o extremismo e apela à tolerância - Terra_:_151\n
Carro visto em rua em que mulher foi encontrada degolada é investigado - Globo.com_:_150\n
Casal real abre processo contra revista por fotos de topless - Terra Brasil_:_160\n
Classe média argentina protagoniza maior panelaço do 2º mandato de ... - Opera Mundi_:_144\n
Com dores abdominais, Roberto Jefferson é internado no RJ - Terra Brasil_:_151\n
Expansão do emprego explica crescimento de gastos das famílias com ... - Terra_:_160\n
Governo seguirá tomando medidas sobre o câmbio, diz Mantega - Reuters Brasil_:_134\n
Polícia apreende metralhadora capaz de derrubar helicóptero no ... - R7_:_148\n
```

em que `\n` significa uma quebra de linha e _ um espaço em branco.

Você pode montar seus próprios casos de teste em diferentes portais de notícia, descobrindo o IP executando ping e abrindo a página da notícia. Para saber mais pesquise sobre arquivos XML e seu uso em distribuição de notícias usando as definições do tipo RSS (Rich Site Summary).

## Informações importantes (LEIA COM ATENÇÃO)

- Sobre a avaliação

    1. Um dos objetivos da disciplina de ICC2 é o **aprendizado individual** dos conceitos de programação. A principal evidência desse aprendizado está nos trabalhos, que são individuais neste curso. Você deverá desenvolver seu trabalho sem copiar trechos de código de outros alunos, nem codificar em conjunto. Portanto, compartilhem idéias, soluções, modos de resolver o problema, mas não o código.
        - O plágio vai contra o código de ética da USP.
        - Quando autores e copiadores combinam, estão ludibriando o sistema de avaliação.
        - O trabalho em grupo e a cooperação entre colegas é em geral benéfico e útil ao aprendizado. Para ajudar um colega você pode lhe explicar estratégias e idéiar. Por exemplo, pode explicar que é preciso usar dois loops para processar os dados, ou que para poupar memória basta usar uma certa estrutura de dados, etc. O que você **não** deve fazer é mostrar o seu código. Mostrar/compartilhar o código pode prejudicar o aprendizado do seu colega:
            - depois de o seu colega ter visto o seu código, será muito mais difícil para ele imaginar uma solução original e própria;
            - o seu colega não entenderá realmente o problema: a compreensão passa pela

prática da codificação e não pela imitação/cópia.
    - Um colega que tenha visto a sua solução pode eventualmente divulgá-la a outros colegas, deixando você numa situação muito complicada, por tabela.
    - O texto acima foi baseado e adaptado da página http://www.ime.usp.br/~mac2166/plagio/, da qual recomendo a leitura completa.

2. Todos os códigos fontes serão comparados por um (ou mais) sistema(s) de detecção de plágio, e **os trabalhos com alta similaridade detectada terão suas notas zeradas**, tanto aqueles relativos ao código de origem quanto do código copiado.

3. A avaliação incluirá a porcentagem de acertos verificada pelo SQTPM e também a análise do seu código, incluindo endentação, comentários, bom uso da memória e práticas de programação. Portanto faça seu código com cuidado, da melhor forma possível.

- Sobre o sistema de submissão (SQTPM):

1. Seu código deverá ser submetido num arquivo fonte .c. Esse arquivo deverá **obrigatoriamente** conter no início um comentário com seu nome, número USP, turma e data da entrega do trabalho.

2. **A data/hora de entrega do trabalho é aquela estipulada no sistema SQTPM. Não haverá adiamento de data/hora. Trabalhos entregues por email NÃO serão aceitos, mesmo que dentro da data/hora estipulada. Faça seu trabalho com antecedência para evitar entregar em cima da hora e ter problemas de submissão, pois o sistema tende a ficar lento com as múltiplas submissões feitas geralmente próximas ao fechamento do sistema.**
    - A submissão é de responsabilidade do aluno, e os problemas comuns à entrega próxima ao fechamento do sistema também. Portanto: problemas de acesso à rede e ao servidor SQTPM **não** serão aceitos como desculpa para entrega por email ou fora do prazo.

3. A compilação do código é feita pelo comando:
```
gcc -o prog proc.c -lm -Wall
```

4. A saída do seu programa deve ser exatamente igual à saída esperada, incluindo: espaços em branco, quebras de linha e precisão decimal.

5. Há um limite de 5 segundos para a execução dos casos de teste e um limite de 16.5MB de memória total para ser utilizada. Você deverá gerenciar bem o tempo de execução e o espaço ocupado para que seu programa fique dentro desses limites, para evitar uso excessivo, pois o sistema irá invalidar o caso em que o limite foi excedido.

6. Ao enviar, aguarde um pouco mais de 10 segundos, **sem recarregar a página nem pressionar ESC**, para obter a resposta. Caso demore mais do que 1 minuto para dar uma resposta, feche e abra novamente a página do servidor. Verifique se seu programa tem algum problema de entrada de dados (ou se tem algum loop infinito ou parada para

pressionamento de tecla). Caso não tenha, aguarde algum 5 minutos e tente novamente.

7. O erro de “Violação de Memória” significa acesso indevido a arranjo ou arquivo.
    - Exemplo 1 de violação de memória:

```
char *array = (char *)malloc(sizeof(char) * N);
scanf("%s", &filename); // acesso indevido, violacao de memoria
free(array);
```

    - Exemplo 2 de violação de memória:

```
int **mat = (int **)malloc(sizeof(int *) * 3);
mat[0] = (int *)malloc(sizeof(int)*10);
for (i = 1; i < 3; i++) {
      // apenas a posicao 0 de mat foi alocada as outras nao
      // portanto esta liberando regiao nao alocada
      // gerando violacao de memoria
      free(mat[i]);
}
```