

Disciplina: Sistema Operacionais Embarcados **Código:** 120961 **Turma:** A

Professor: Diogo Caetano Garcia

Aluno/Matrícula: Fábio Barbosa Pinto – 11/0116356

Questionário: 10_Clientes_HTTP_2

1.Utilize o programa criado pelo professor para baixar as páginas principais dos seguintes sites:

(a) www.google.com

Abrindo o socket para o cliente... Feito! Obtendo o IP do servidor... Feito! Conectando o socket ao IP 172.217.30.100 pela porta 80... Feito! Pedido HTTP:

GET / HTTP/1.1 Host: www.google.com User-Agent: HTMLGET 1.1 Accept: /

(b) www.google.com.br

Abrindo o socket para o cliente... Feito! Obtendo o IP do servidor... Feito! Conectando o socket ao IP 172.217.28.131 pela porta 80... Feito! Pedido HTTP:

GET / HTTP/1.1 Host: www.google.com.br User-Agent: HTMLGET 1.1 Accept: /

(c) www.unb.br

Abrindo o socket para o cliente... Feito! Obtendo o IP do servidor... Feito! Conectando o socket ao IP 164.41.102.70 pela porta 80... Feito! Pedido HTTP:

GET / HTTP/1.1 Host: www.unb.br User-Agent: HTMLGET 1.1 Accept: /

(d) fga.unb.br

Abrindo o socket para o cliente... Feito! Obtendo o IP do servidor... Feito! Conectando o socket ao IP 164.41.86.15 pela porta 80... Feito! Pedido HTTP:

GET / HTTP/1.1 Host: www.fga.unb.br User-Agent: HTMLGET 1.1 Accept: / Comente os resultados obtidos para cada página, em termos das respostas HTTP e HTML obtidas.

Código

```
#include <fcntl.h> #include <unistd.h> #include <stdio.h> #include <sys/socket.h> #include  
<arpa/inet.h> #include <stdlib.h> #include <netdb.h> #include <string.h>
```

```
char *build_get_query(char *host, char *page); char *get_ip(char *host);
```

```
int main(int argc, char **argv) { struct sockaddr_in servidorAddr; int socket_id; int port = 80;  
char *host, *ip, *get, *page, buf[BUFSIZ+1]; FILE *fp;
```

```
if(argc != 3)
```

```
{
    fprintf(stderr, "Uso: %s host pagina\n", argv[0]);
    fprintf(stderr, "  host: o endereco do website. ex: www.unb.br\n");
    fprintf(stderr, "  pagina: a pagina para obter. ex: /\n");
    exit(2);
}
host = argv[1];
page = argv[2];

fprintf(stderr, "Abrindo o socket para o cliente... ");
if((socket_id = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0)
{
    fprintf(stderr, "Erro na criacao do socket!\n");
    exit(0);
}
fprintf(stderr, "Feito!\n");

fprintf(stderr, "Obtendo o IP do servidor... ");
ip = get_ip(host);
fprintf(stderr, "Feito!\n");

fprintf(stderr, "Conectando o socket ao IP %s pela porta %d... ", ip, port);
memset(&servidorAddr, 0, sizeof(servidorAddr));
servidorAddr.sin_family = AF_INET;
servidorAddr.sin_addr.s_addr = inet_addr(ip);
servidorAddr.sin_port = htons(port);
if(connect(socket_id, (struct sockaddr *) &servidorAddr,
           sizeof(servidorAddr)) < 0)
{
    fprintf(stderr, "Erro na conexao!\n");
    exit(0);
}
fprintf(stderr, "Feito!\n");

get = build_get_query(host, page);
fprintf(stderr, "Pedido HTTP:\n\n");
fprintf(stderr, "-----\n");
fprintf(stderr, "%s", get);
fprintf(stderr, "-----\n");

fprintf(stderr, "Enviando o pedido HTTP ao servidor... ");
write(socket_id, get, strlen(get));
fprintf(stderr, "Feito!\n");

free(get);
free(ip);
```

```
fprintf(stderr, "Recebendo o resultado HTML e o escrevendo no arquivo 'saida.html'... ");
fp = fopen("saida.html", "w");
int htmlstart = 0, tmpres;
char * htmlcontent;
while((tmpres = read(socket_id, buf, BUFSIZ)) > 0)
{
    buf[tmpres] = '\0';
    if(htmlstart == 0)
    {
        // Under certain conditions this will not work.
        // If the \r\n\r\n part is split into two messages
        // it will fail to detect the beginning of HTML content
        htmlcontent = strstr(buf, "\r\n\r\n");
        if(htmlcontent != NULL)
        {
            htmlstart = 1;
            htmlcontent += 4;
        }
    }
    else htmlcontent = buf;

    if(htmlstart) fprintf(fp, "%s", htmlcontent);
}
if(tmpres < 0)
    fprintf(stderr, "Erro no recebimento de dados!\n");
fprintf(stderr, "Feito!\n");
close(socket_id);
fclose(fp);
return 0;
}

char *build_get_query(char *host, char *page) { char *query; char *getpage = page; char *tpl
= "GET %s HTTP/1.1\r\nHost: %s\r\nUser-Agent: HTMLGET 1.1\r\nAccept: /\r\n\r\n"; query = (char
*)malloc(strlen(host)+strlen(getpage)+strlen(tpl)); sprintf(query, tpl, getpage, host); return query; }

char *get_ip(char *host) { struct hostent *hent; int iplen = 15; //XXX.XXX.XXX.XXX char *ip =
(char *)malloc(iplen+1); memset(ip, 0, iplen+1); if((hent = gethostbyname(host)) == NULL) {
herror("Can't get IP"); exit(1); } if(inet_ntop(AF_INET, (void *)hent->h_addr_list[0], ip, iplen) == NULL)
{ perror("Can't resolve host"); exit(1); } return ip; }
```

Como previsto todos tem um endereço externo diferente, porem o mais interessante são as rotas desses endereços. Pode-se verificar que existe um grau de variação em apenas 2 campos em cada exemplo, isso mostra que mesmo sendo endereços diferentes em algum momento no caminho existiu um ponto em comum que poderia nos levar até o google.com e ao google.com.br, assim como os servidores da unb que até o ponto 164.41, que é o caminho em comum entre o site da unb e o site da fga.