MAC122 – Princípios de Desenvolvimento de Algoritmos
Segundo Semestre de 2014 – BMAC – IMEUSP – Prof. Marcilio
EXERCÍCIO PROGRAMA II – Entregar 11/Out/2013

O programa recebe mensagens da rede.
As mensagens podem ser de novas notícias recebidas ou solicitação de usuários para mostrar as notícias dentro de um intervalo de tempo de um dia.

Notícias recebidas

As notícias são recebidas no formato texto. No início está a hora em que foi gerada:

hh:mm:ss <conteúdo>

Exemplos:

```
12:10:33 bla bla bla . . . . .
03:00:02 tla tla tla . . . . .
```

As notícias não chegam necessariamente em ordem cronológica. O objetivo do programa é manter as notícias ordenadas cronologicamente para permitir uma consulta rápida. Para isso, o programa deve manter uma lista ligada com as notícias nesta ordem.

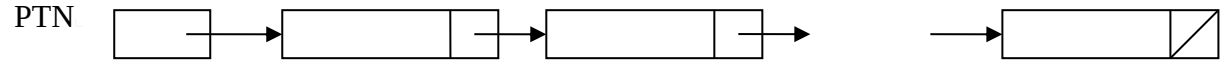
Cada elemento da lista ligada contém:

```
struct Lista {
  int hora; /* melhor guardar o horário convertido para inteiro hhmmss
               para facilitar a comparação */
  char * Texto; /* ponteiro para o texto da notícia */
  Noticia PN; /* ponteiro para a próxima notícia */
}

typedef struct Lista * Noticia;
```

O texto das notícias é uma cadeia de caracteres e tem um tamanho variável. Por isso temos um ponteiro para uma área de memória que contém esse texto em vez do próprio texto que estará numa área de memória alocada dinamicamente.

Lista Ligada de Notícias

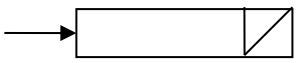


Solicitação de usuários

As solicitações de usuário chegam também como texto no formato:

hh:mm:ss ou **hh:mm** ou **hh**

Ou seja, quando não há texto, é mensagem de solicitação de usuário.



Exemplos:

14:23:45 – Responde com a primeira notícia que chegou no horário maior ou igual a **14:23:45**
02:34 - Responde com a primeira notícia que chegou no horário maior ou igual a **02:34:00**
23 - Responde com a primeira notícia que chegou no horário maior ou igual a **23:00:00**

Simulação da rede

Vamos simular a chegada de mensagens da rede através de uma função **GeraMsg(char msg[])** que gera mensagens de notícias recebidas ou de solicitação de usuários. Veja abaixo a função.

A resposta de mensagens de solicitação de usuário é simulada mostrando a mensagen no vídeo.

Gravação das mensagens recebidas

As notícias recebidas são mantidas em memória (lista ligada) para acesso rápido quando há uma solicitação de usuário, mas devem ser gravadas em disco num arquivo de texto **"arqnoticias.txt"** na ordem de chegada.

Também gravar num outro arquivo de texto, **"arqusuario.txt"**, as notícias solicitadas pelos usuários.

Estrutura do programa principal

O programa fica em loop infinito. Só termina quando for digitado **<ctrl>c**.
A estrutura fica então:

```
while (1) {
    GeraMsg(m);
    Se m é solicitação do usuário {
             procure na lista ligada a notícia solicitada
             mostre no vídeo a notícia
             grave a notícia em arqusuario.txt
    }
    Se m é notícia recebida {
             inserir a notícia na posição adequada da lista ligada
             grave a notícia em arqnoticias.txt
    }
}
```

Função GeraMsg

A seguir a função que gera as notícias recebidas e as solicitações de usuário.

Se tiver dúvida do funcionamento, rode o programa principal abaixo que esclarece o formato das mensagens.

```
#include <string.h>
#include<stdio.h>
#include<stdlib.h>
#include <time.h>
```

```
    static char n1[20][15] = {"Lucas", "Caminhao", "Aviao", "Passaro", "Maquina",
"Carro", "Carruagem", "Cachorro", "Cabrito", "Olaria", "Jornal", "Filho", "Pai",
"Avo", "Alfinete", "Agulha", "Sapato","Rua","Livro","Quadro"};
    static char n2[20][15] = {"corre", "voa", "anda", "mexe", "capota", "segura",
"esculhamba", "vira", "separa", "mata", "recebe", "pavoneia", "se tranca", "bebe",
"lota", "celebra", "cheira","move","abre","lubrifica"};
    static char n3[20][15] = {"rapido","baixo", "o terno", "a camisa", "o sapato",
"dormindo", "o telefone", "alto", "chapado", "a faca", "tonto", "lento", "grande",
"bonito", "pequeno", "feio","barato", "chorando","lotado", "vazio"};
    static char f[20][10] =
{"UPI","UOL","IG","GLOBO","SBT","GAZETA","RNT","RPRESS","EURO","SCAND","IBERO","FR
AN","KVT","LGO","KRIM","FA","FB","FC","FD","FE"};

void GeraMsgUsuario (char msgu[]) {
     int k = rand() % 3;
     /* horario */
     switch (k) {
       case 2: sprintf(msgu, "%02d:%02d:%02d", rand() % 24, rand() % 60, rand() %
60); break;
       case 1: sprintf(msgu, "%02d:%02d", rand() % 24, rand() % 60); break;
       case 0: sprintf(msgu, "%02d", rand() % 24); break;
     }
}

void GeraNoticia (char nm[]) {
    int i,j,k,l; char lf[]={10,0}; char linha[]={"*linha*00X\n"}; char bla[]={"
blablabla"};
    /* horario e <agencia>" */
    sprintf(nm, "%02d:%02d:%02d ", rand() % 24, rand() % 60, rand() % 60);
    strcat(nm, f[rand()%20]);
    strcat(nm,lf);
    /* Manchete da notícia - 3 palavras */
    /* primeira palavra */
    i = rand() % 20;
    strcat(nm, n1[i]);
    strcat(nm, " ");
    /* segunda palavra */
    i = rand() % 20;
    strcat(nm, n2[i]);
    strcat(nm, " ");
    /* terceira palavra */
    i = rand() % 20;
    strcat(nm, n3[i]);
    strcat(nm, " ");
    /* proximas linhas */
    strcat(nm, lf);
    j=rand()%10; /* ate 10 linhas */
    for (i = 0; i <= j; i++) {
        k = rand() % 6; linha[8]='0' + i / 10; linha[9] = '0' + i % 10;
        for(l = 0; l <= k; l++) strcat(nm,bla);
        strcat(nm,lf);
    }
}

/* gera msg de notícia recebida ou de solicitação de usuário */
void GeraMsg(char msg[]) {
     msg[0] = 0; /* string vazio */
     int k = rand() % 10;
```

```
      /* espera 1 segundo */
      sleep(1000);
      if (k == 0) GeraMsgUsuario (msg);
      else GeraNoticia (msg);
}

int main() {
   char msg[2000];
   srand(76381);
   while (1) {
         GeraMsg(msg);
         /* mostra notícia */
         printf("%s\n\n",msg);
   }
  system("PAUSE");
  return 0;
}
```

Gravação em disco

Para gravar o arquivo:

```
   FILE *f;
   char texto[1000];
   ...
   /* abrir o arquivo */
   f = fopen("arqnoticias.txt", "w"); ou
   f = fopen("arqusuario.txt", "w");
   ...
   /* para gravar um texto qualquer - texto é uma cadeia de caracteres */
   fputs(texto, f);
   ...
   /* fechar o arquivo no final */
   fclose(f);
```