

MAC122 – Princípios de Desenvolvimento de Algoritmos
Segundo Semestre de 2013 – BMAC – IMEUSP – Prof. Marcilio
EXERCÍCIO PROGRAMA II – Entregar 27/Out/2013

O programa recebe mensagens da rede.

As mensagens podem ser de novas notícias recebidas de agências geradoras ou solicitação de usuários para mostrar as notícias que atendem a determinado critério.

Notícias recebidas

As notícias são recebidas no formato texto, contendo o nome da fonte geradora (agência):

<agência> <conteúdo>

O objetivo do programa é manter as notícias ordenadas por <agência> para permitir uma consulta rápida. Para isso, o programa deve manter uma lista ligada com as notícias na ordem de chegada. Para tornar o processo mais eficiente em vez de uma só lista, vamos manter N listas, uma para cada <agência>. Pode ser N < 50. Assim, cada notícia que chega, é inserida no início da lista da <agência>. O separador do nome da agência e do conteúdo é um caractere LF (line-feed - \10 ou \n).

Cada elemento da lista ligada contém:

```
struct Lista {
    char * Texto; /* ponteiro para o texto da notícia */
    Noticia PN; /* ponteiro para a próxima notícia desta hora */
}
```

```
typedef struct Lista * Noticia;
```

O texto das notícias 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.

Temos que ter uma tabela com os nomes das agências que geraram notícias:

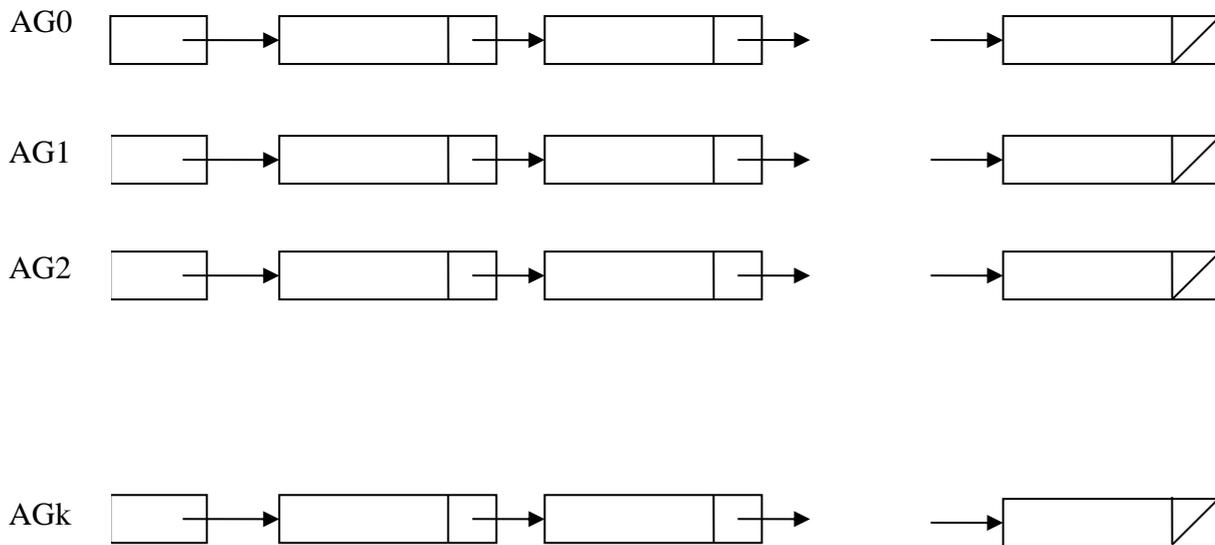
char TABAG[50][10] – no máximo 50 agências – nomes de no máximo 10 caracteres.

Quando chega uma notícia, deve-se procurar o nome da agência nesta tabela. Se não estiver, inserir na primeira posição livre.

Paralela a esta tabela tem a tabela de ponteiros para cada uma das listas.

```
Noticia PTH[50]; /* noticias */
```

TABAG PTH Listas



Solicitação de usuários

As solicitações de usuário chegam também como texto no formato:

<nn> <agência>

O separador do número e do nome da agência é um branco.

Exemplos:

05 RPRESS

Devolve as 5 últimas notícias entre da agência RPRESS.

00 GAZETA

Devolve todas as notícias entre da agência GAZETA.

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 devolução de mensagens de solicitação de usuário é simulada mostrando as mensagens no vídeo.

Gravação das mensagens recebidas

As notícias recebidas são mantidas em memória (lista ligada) para acesso rápido quando há uma solitação de usuário, mas devem ser gravadas em disco num arquivo de texto **"arqnoticias.txt"**. A primeira linha de cada notícia deve ter a data e hora em que a mesma chegou.

Também gravar num outro arquivo de texto as solicitações de usuário, "**arquuario.txt**" também com a data e hora em que foram solicitadas. Não precisa gravar as mensagens que foram mostradas ao usuário.

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 {
        mostre no vídeo as notícias solicitadas
        grave a solicitação
    }
    Se m é notícia recebida {
        inserir a data/hora no inicio da notícia
        inserir a notícia na lista ligada da agencia
        grave a notícia
    }
}
```

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] = {"Lula", "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;
    /* quantidade de notícias */
    sprintf(msgu, "%02d ", rand() % 5);
    /* agência */
    strcat(msgu, f[rand() % 20]);
}

void GeraNoticia (char nm[]) {
```

```

int i,j,k,l; char lf[]={10,0}; char linha[]={"*linha*00X\n"}; char bla[]={"
blablabla"};
/* cabeçalho da noticia <agencia>" */
strcpy(nm, f[rand()%20]);
strncat(nm,lf,1);
/* Manchetete da notícia - 3 palavras */
/* primeira palavra */
i = rand() % 20;
strncat(nm, n1[i], 15);
strncat(nm, " ", 1);
/* segunda palavra */
i = rand() % 20;
strncat(nm, n2[i], 15);
strncat(nm, " ", 1);
/* terceira palavra */
i = rand() % 20;
strncat(nm, n3[i], 15);
strncat(nm, " ", 1);
/* proximas linhas */
strncat(nm, lf, 1);
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++) strncat(nm,bla,10);
    strncat(nm,lf,1);
}
}

/* 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 o horário */
        time_t now = time(0);
        /* converte now para string */
        char* dt = ctime(&now);
        /* mostra data-hora */
        printf("%s", dt);
        /* 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 */
fputs(texto, f);
...
/* fechar o arquivo no final */
fclose(f);
```