

Trabalho Individual

Coleta de velocidades de ônibus
no Rio de Janeiro

Velocidade de ônibus no Rio de Janeiro

- A prefeitura do Rio de Janeiro fornece dados GPS, em tempo real, das linhas de ônibus
 - Requisições HTTP buscam informações sobre uma determinada linha de ônibus

Campo	Descrição
DataHora	Data e hora da coleta do dado
Ordem	Identificação encontrada na lateral do ônibus
Linha	Linha do ônibus
Latitude	Latitude na coleta
Longitude	Longitude na coleta
Velocidade	Velocidade da coleta
Direção	Graus em relação ao norte

Informações do ônibus

- Informações são coletadas por HTTP no servidor da prefeitura
 - Ex: linha 474
 - Requisição: <http://dadosabertos.rio.rj.gov.br/apiTransporte/apresentacao/rest/index.cfm/obterPosicoesDaLinha/474>
 - Resposta

```
{"COLUMNS": ["DATAHORA", "ORDEM", "LINHA", "LATITUDE", "LONGITUDE", "VELOCIDADE"], "DATA": [{"09-29-2016 09:20:39", "A29137", 474.0, -22.868481, -43.293209, 20.0}, {"09-29-2016 13:39:30", "A29008", 474.0, -22.95159, -43.180988, 27.0}, {"09-29-2016 13:41:54", "A29133", 474.0, -22.929541, -43.186958, 57.0}, {"09-29-2016 13:46:27", "A29153", 474.0, -22.93058, -43.186001, 58.0}, {"09-29-2016 13:46:30", "A29155", 474.0, -22.916269, -43.19421, 60.0}, {"09-29-2016 13:48:28", "A29159", 474.0, -22.89426, -43.26321, 0.0}, {"09-29-2016 13:49:58", "A29138", 474.0, -22.8946, -43.262192, 16.0}, {"09-29-2016 13:50:18", "A29130", 474.0, -22.89427, -43.263069, 0.0}, {"09-29-2016 13:57:03", "A29141", 474.0, -22.986759, -43.190868, 15.0}, {"09-29-2016 13:57:14", "A29165", 474.0, -22.9575, -43.17733, 30.0}, {"09-29-2016 13:57:45", "A29154", 474.0, -22.90028, -43.225029, 18.0}, {"09-29-2016 13:58:18", "A29003", 474.0, -22.905649, -43.214081, 14.0}, {"09-29-2016 13:58:39", "A29013", 474.0, -22.907709, -43.209999, 26.0}, {"09-29-2016 13:58:40", "A29001", 474.0, -22.900021, -43.223389, 23.0}, {"09-29-2016 13:58:41", "A29117", 474.0, -22.95369, -43.181499, 18.0}, {"09-29-2016 13:58:43", "A29157", 474.0, -22.98637, -43.191021, 28.0}, {"09-29-2016 13:58:45", "A29015", 474.0, -22.963671, -43.174068, 16.0}, {"09-29-2016 13:58:46", "A29079", 474.0, -22.896761, -43.257011, 19.0}, {"09-29-2016 13:58:46", "A29173", 474.0, -22.908621, -43.19693, 18.0}, {"09-29-2016 13:58:50", "A29012", 474.0, -22.90913, -43.204411, 16.0}, {"09-29-2016 13:58:54", "A29049", 474.0, -22.966841, -43.180641, 33.0}, {"09-29-2016 13:58:54", "A29113", 474.0, -22.971161, -43.189419, 39.0}, {"09-29-2016 13:58:55", "A29183", 474.0, -22.905331, -43.210159, 35.0}, {"09-29-2016 13:59:07", "A29131", 474.0, -22.972321, -43.187599, 29.0}]}
```

Trabalho

- Coletar, durante T segundos, a velocidade de N ônibus escolhidos pelo usuário
 - Ao final do programa, imprimir todas as informações coletadas
 - Imprimir informação da linha do ônibus com maior velocidade média no período
- Início
 - Programa pede para o usuário um número " N " de ônibus a serem coletados e um tempo " T " de coleta (em segundos)
 - Após isso, o programa pede a identificação (ordem) de cada ônibus a ser coletado
 - Cria-se uma vetor (dinâmico) com a ordem de cada ônibus

Trabalho

- Programa coleta, a cada segundo, a velocidade de todos os "n" ônibus e insere numa matriz
 - Linha da matriz representa um ônibus
 - Colunas da matriz representam velocidade a cada segundo
 - Ou seja, realiza os seguintes passos:
 - Pede sequencialmente para o servidor da prefeitura a velocidade dos 'n' ônibus, inserindo na matriz a velocidade de cada ônibus
 - Espera 1 segundo e realiza o passo anterior, até o número de segundos ser igual ao tempo de coleta
- Uso da matriz é obrigatório! Não imprimam a velocidade coletada imediatamente após a coleta
 - Matriz dinâmica

Trabalho

- Obviamente, tempo de coleta do programa será bem maior do que o especificado pelo usuário, pois existe um tempo para realizar a requisição para cada ônibus
 - Não se preocupem com isso!

Trabalho

- No final da coleta, programa deverá
 - Imprimir informações da matriz, com a ordem de cada ônibus e cada velocidade coletada
 - Ou seja, ler informações da matriz e usar o vetor com os nomes dos ônibus para fazer a correspondência entre linhas da matriz e nomes dos ônibus
 - Imprimir nome e linha do ônibus que obteve a maior velocidade média do período
 - Já veremos como obter a linha do ônibus
- Ex: com 4 segundos de coleta e 2 ônibus:
A29136: 60 km/h ; 50 km/h ; 20 km/h ; 40 Km/h
A29165: 40 km/h ; 50 km/h ; 30 km/h ; 60 Km/h
Linha com maior velocidade: 474

Função utilizada

- Função que retorna um ponteiro para uma estrutura de um ônibus específico (mais infos no próximo slide)

```
Onibus* infoOnibus (char* ordemOnibus)
```

- Função disponível nos arquivos dadosOnibus.h e dadosOnibus.c

Estrutura de um ônibus

- Retornada pela função infoOnibus
 - Na verdade é um ponteiro para uma estrutura desse tipo, alocada na própria infoOnibus (má prática de programação do professor)
 - Reparem que a função infoOnibus permite obter a velocidade e a linha do ônibus

```
typedef struct onibus {  
    char data[24];  
    char ordem[8];  
    char linha[8];  
    double latitude;  
    double longitude;  
    float velocidade;  
    struct onibus* prox;  
    struct onibus* ant;  
}Onibus;
```

Estrutura de um ônibus

- Retornada pela função infoOnibus
 - Na verdade é um ponteiro para uma estrutura desse tipo, alocada na própria infoOnibus (má prática de programação do professor)
 - Reparem que a função infoOnibus permite obter a velocidade e a linha do ônibus

```
typedef struct onibus {  
    char data[24];  
    char ordem[8];  
    char linha[8];  
    double latitude;  
    double longitude;  
    float velocidade;  
    struct onibus* prox;  
    struct onibus* ant;  
}Onibus;
```

Não se preocupem
com esses ponteiros.
Não os usaremos!

Requisitos

- Arquivo `curl.exe` deve estar no mesmo diretório que os arquivos `dadosOnibus.c` e `dadosOnibus.h`
- Seu computador deve estar conectado à Internet
- No arquivo fornecido existe um projeto chamado `TesteOnibus`. Compile-o e teste para verificar o funcionamento da biblioteca
 - É MUITO IMPORTANTE QUE ESSE TESTE SEJA FEITO IMEDIATAMENTE

Como encontrar uma identificação (ordem) de ônibus válida

- Use a requisição, substituindo 474 pela linha desejada. Assim, várias ordem de ônibus serão listadas e você poderá escolher uma
 - <http://dadosabertos.rio.rj.gov.br/apiTransporte/apresentacao/rest/index.cfm/obterPosicoesDaLinha/474>
- Ou use o programa de exemplo (o do código testeOnibus.c)
 - Ele pede uma linha e retorna todas as ordens daquela linha circulando no momento

Função sleep

- Função útil para delimitar o tempo de medida
- Utilizada com a inclusão de Windows.h
 - `#include <windows.h>`
- “pausa” a execução do programa em 'x' milissegundos
 - Exemplo de “pausa” de 1 segundo

`Sleep(1000)`

Como começar a fazer

- Preencha o arquivo "testeOnibus.c" com o código do trabalho
 - Código existente já pede para o usuário a ordem e retorna sua velocidade
- Código necessário está no site em bibliotecaDesenvolvidas.zip

Trabalho Individual

- Entrega
 - Código .c enviado por e-mail rodsouzacouto@ieee.org
 - O código realizado pelo(a) aluno(a) pode estar apenas em um .c, sem separação de funções
 - Deve ser entregue em condições de ser compilado e executado
 - Prazo - 06:00 (manhã) de 03/07/2018
- Valor
 - 3 pontos na média dos trabalhos
- Trabalhos copiados receberão ZERO