

Semestrální práce IUJC 2019

Způsob vypracování a odevzdání práce:

- Semestrální práce je vypracována samostatně (není přípustná **duplicita** kódu)
- Student může být v případě potřeby vyzván k obhájení práce, tzn., že student musí být schopen zodpovědět dotazy k práci
- Zápis kódu odpovídá zvykům strukturovanému programování
- Všichni studenti odevzdávají práci do STAGU v sekci odevzdávání prací (Blok: Sem IUJC 2019 - Semestrální práce IUJC 2019)
- **Nejzazší** termín odevzdání práce je **30. 12. 2019 do 20:00** (po tomto termínu nebudou semestrální práce přijímány a student ztrácí možnost získání zápočtu)
- Odevzdávají se pouze **zdrojové** soubory (*.c, *.h), archivovány ve **ZIPu** (ne v .rar, .tar, atd, = hodnocení 0b)
- Použít funkci **_CrtDumpMemoryLeaks()** pro ověření, že veškeré alokované paměťové prostředky byly navraceny zpět.

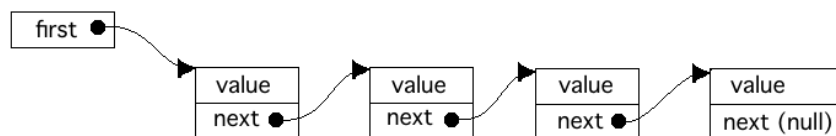
Hlavní idea

Cílem semestrální práce je jednoduchá analýza měření spotřeby vody od jednotlivých senzorů. Data o spotřebě jsou uložena v hodinových intervalech v csv souboru, který má následující hlavičku `id;sensor_id;time;m3`, viz Obrázek 1.

id	sensor_id	time	m3
151054	35	2019-11-16 02:00:00	0.1830
151056	2	2019-11-16 02:00:00	0.0000
151063	41	2019-11-16 02:00:00	0.0000
151064	222	2019-11-16 02:00:00	0.2965
151071	35	2019-11-16 03:00:00	0.1990

Obrázek 1: Ukázka csv dat

Data o měřeních jsou primárně načtena do lineárního seznamu, viz Obrázek 2.



Obrázek 2: koncepce lineárního seznamu

Analýza průměrné spotřeby (v litrech) za jednotlivé dny v týdnu v hodinových intervalech (vždy za 24 hodin), je následně uložena do dynamického 2D pole, viz Obrázek 3.

den/hod	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	154	5	69	103	165	62	35	200	8	155	187	67	71	13	167	89	82	7	43	59	196	156	40	15
1	135	116	167	164	15	77	192	105	197	104	87	167	54	160	161	148	30	76	191	87	193	192	109	66
2	199	132	140	127	185	32	59	45	148	45	97	12	77	147	20	192	55	98	100	171	151	111	98	34
3	164	44	28	53	184	126	13	112	186	185	138	101	64	160	197	74	153	45	86	12	82	162	114	113
4	127	31	15	136	87	147	45	148	134	126	107	3	49	119	96	73	112	54	192	164	71	3	138	40
5	95	181	138	50	116	23	142	99	188	86	134	177	127	14	54	139	32	189	195	159	170	81	4	154
6	43	56	10	41	144	160	36	157	27	45	141	117	56	137	181	79	42	176	136	171	2	200	74	2

Obrázek 3: Příklad 2D matice průměrů (7 dnů x 24 h)

Modul DateTime:

Hlavičkový soubor **datetime.h** obsahuje

datetime.h

```
typedef struct {
    int year;
    int month;
    int day;
    int hour;
    int min;
    int sec;
    int dayInWeek;
}tDateTime;
```

a deklaraci funkcí:

```
tDateTime dejDateTime(char* datetime) // převádí vstup (2018-05-01 01:00:00)
na strukturu, využívá dejDenVTydnu
int dejDenVTydnu(int r, int m, int d) //vrací den v týdnu (0-PO,...,6-NE)
*výpočet viz poznámka
```

soubor **datetime.c** obsahuje pouze implementace (definici) těchto funkcí

Modul mereni:

Hlavičkový soubor **mereni.h** obsahuje

mereni.h

```
typedef struct mereni{
    int idMereni;
    tDateTime timestamp;
    float m3;
    struct mereni * dalsi;
}tMereni;
```

a deklaraci funkcí:

```
tMereni * vytvorMereni(int idMereni, tDateTime ts,float m3) // vrací adresu
dynamické alokace mereni, ukazatel dalsi se nastaví na NULL
void vypisMereni (tMereni *mereni) //vypíše mereni na obrazovku
```

soubor **mereni.c** obsahuje pouze implementace (definici) těchto funkcí

Modul seznamMereni:

Hlavičkový soubor **seznamMereni.h** obsahuje

```
typedef struct {
    int idSenzoru;
    tDateTime datumOd;
    tDateTime datumDo;
    tMereni *seznam;
}tSeznamMereni;
```

deklaraci funkcí:

```
int *dejPoleIdSenzoru() // vrací pole id všech senzorů v souboru
tSeznamMereni *nactiMereni(int idSenzoru) // načte všechna měření daného
senzoru, využívá vytvorMereni, pokud již nějaký seznam existuje, musí být
dealokován
void vypisMereni // vypíše všechna měření, využívá vypisMereni
tMereni *odeberMereni(DateTime timestamp) // odebere a vrátí požadované měření
ze seznamu
void dealokujSeznam() // smaž (dealokuj) všechny záznamy a tSeznam
float **analyzuj(DateTime datumOd, DateTime datumDo) // Výpočet průměru pro
všechny kombinace den v týdnu, který vrátí dynamické 2D pole 7x24 „den v
týdnu“ x „hodina dne“ (v litrech)
void dealokujMatici() // smaž (dealokuj) matici
float dejOdchylku(DateTime timestamp) () //vrátí odchylku v litrech daného
záznamu (měření v dle timestamp) oproti správnému průměru z matice průměrů
```

soubor **seznamMereni.c** obsahuje pouze implementace (definici) těchto funkcí

Modul main:

Hlavní soubor **main.c** obsahuje textové menu:

1. Vypiš id všech senzorů - využívá `dejPoleIdSenzoru`
2. Načti měření – využívá `nactiMerei`
3. Vypiš načtená měření - využívá `vypisMereni`
4. Odeber měření - využívá `odeberMereni`
5. Vypiš analýzu průměrů – využívá `analyzuj`
6. Ulož analýzu průměrů do souboru CSV – využívá `analyzuj`
7. Vypiš odchylku – využívá `dejOdchylku`
8. Dealokuj matici – využívá `dealokujMatici`
9. Smaž měření využívá `dealokujSeznam`

Pro realizaci funkcionalit menu, lze vytvářet vlastní pomocné funkce.

*) <https://kalendar.beda.cz/jak-jednoduse-zjistit-den-v-tydnu-pro-dane-datum>