

# Cvičenie 10

# Obsah

- organizačné záležitosti
- vstupy & výstupy
- úloha 10

# Organizačné záležitosti

- úloha 7 (vlastná) – uzavretá
- úloha 6 (union-find) - opravená
  - bodovanie (efektivita)
  - problémy
    - podcenenie náročnosti
    - nefunkčný `get_count(void)`
    - chybná voľba reprezentanta
  - odporúčania
    - komentujte
    - inicializačné listy

# Organizačné záležitosti

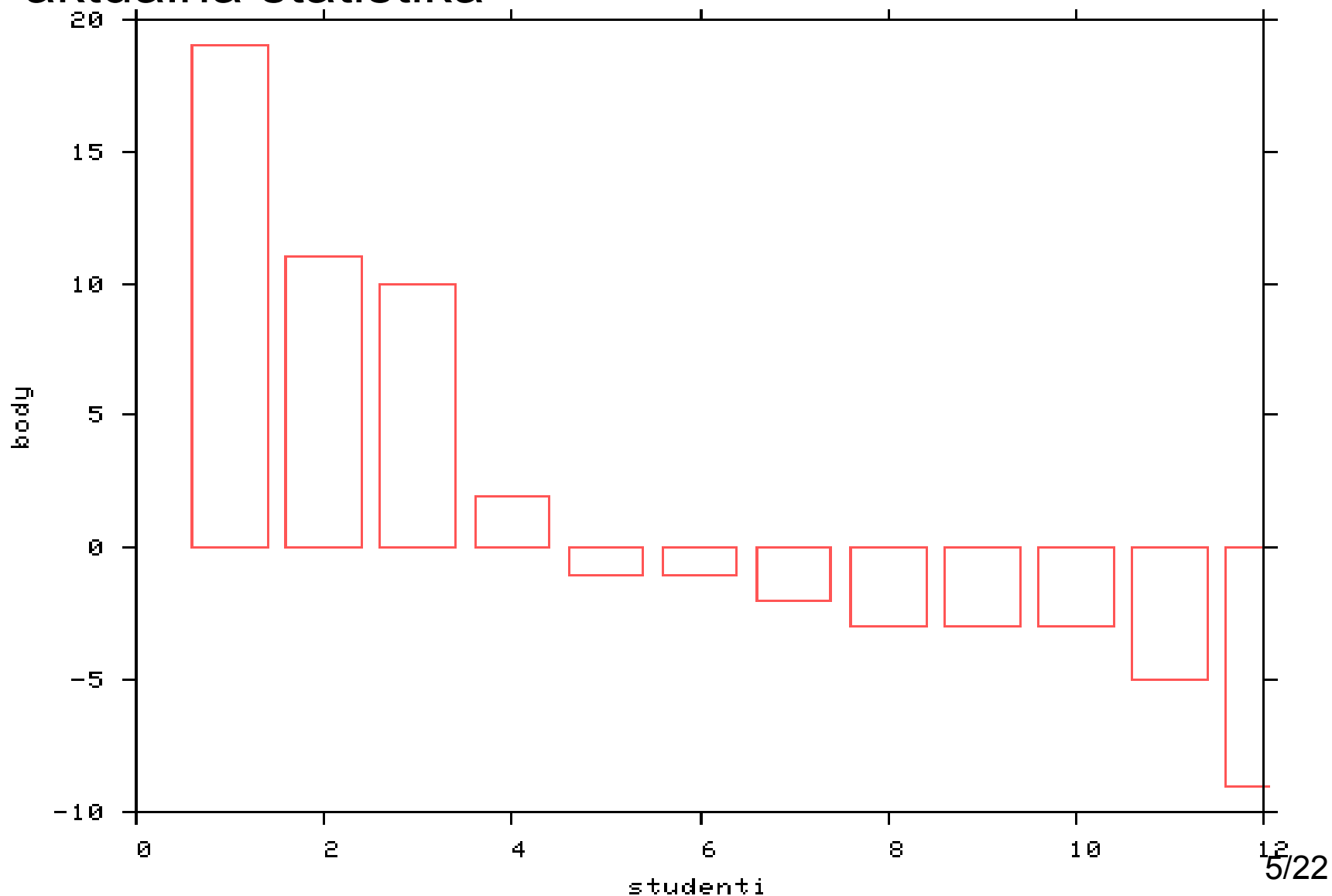
- úloha 8 (agenti)
  - do dnes
  - bonus za ľahko rozšíriteľnú hierarchiu, modifikáciu prístupu
- úloha 9 (120)
  - deadline o týždeň
  - dodatočné rozšírenie o bonus (vid' DF)
    - +1 bod ak opravíte súbor neukončený “\n”
    - ak ste už odovzdali, pôvodný súbor nechajte (overenie odovzdania), pridajte súbor `stodvacet_bonus.cc` a pošlite mail

# Organizačné záležitosti

- úloha 10
  - bude zadaná v závere
  - posledná povinná :)
- plán do konca semestra
  - ostávajú 3 cvičenia
  - výnimky
  - šablony
  - diskusia neurobených úloh, záverečného príkladu
  - príp. predstavenie nejakých užitočných nástrojov

# Organizačné záležitosti

- aktuálna štatistika



# Vstupy a výstupy v C - opakovanie

- otvorenie a zatvorenie súboru

```
FILE *fp;
```

```
fp = fopen(názov,režim); // pri chybe vracia NULL
```

```
fclose(fp); // pri chybe vracia EOF
```

- formátovaný IO

- printf(formát,výrazy); // vracia počet znakov

- scanf(formát,ukazatele); // vracia počet položiek

- fprintf(fp,formát,výrazy);

- fscanf(fp,formát,ukazatele);

- sprintf(buffer,formát,výrazy);

- sscanf(buffer,formát,ukazatele);

# Vstupy a výstupy v C - opakovanie

- znakový IO
  - `getc()`, `putc()`, `getchar()`, `putchar()`, `ungetc()`, `gets()`, `puts()`...
- ďalšie funkcie
  - `ferror(fp)`, `feof(fp)`, `fflush(fp)`, `clearerr(fp)`...
- viac info

<http://www.fi.muni.cz/usr/jkucera/pb071/sl8.htm>



# Vstupy a výstupy

- prúdový prístup
- štandard (cin, cout, cerr), operátory (<<, >>)
- ponúkané je oveľa viac :)
- všetko je v namespace std
- objektovo riešené → výhody:
  - nástroje súčasťou knižníc, hierarchie objektov
  - bezpečnosť
  - jednoduchosť použitia
  - rozširiteľnosť
  - jednotnosť
  - preťažovanie operátorov

# Vstupy a výstupy - operátory

- preťažené na rôzne typy (na výstup aj výrazy)
- vracajú svoj ľavý operand → reťazenie
- ak operácia neúspešná, vracia ekvivalent false

# Metódy na prúdoch (prehľadovo)

- vstup

- `get()`, `getline()`, `ignore()`, `read()`, `tellg()` - vracia pozíciu v súbore, `seekg()` - presunutie v súbore, `putback()` - vsunutie znaku do vstupného bufferu, `unget()` - posledný znak do vstupného bufferu, `peek()` - nahliadnutie na ďalší znak na vstupe, `sync()` - zmaže buffer

- výstup

- `put()`, `write()`, `flush()` - vyprázdnenie bufferu, `tellp()`, `seekp()`

# Metódy na prúdoch

- pozn. o getline()
  - varianta A (<string>)
    - istream& getline(istream& is, string& s, char delimiter = '\n');
    - prečíta riadok z “is” a uloží do “s”
    - pracuje s reťazcami (string)
  - varianta B (<fstream>)
    - istream& getline(char\* buffer, streamsize num, char delim);
    - prečíta num-1 znakov z prúdu, na ktorý aplikovaná a uloží do “buffer”
    - pracuje s char\*
    - oddeľovač nepovinný

# Manipulátory (prehľadovo)

- `<iomanip>`
- vid' cvicenie05.pdf
- menia správanie prúdov
  - väčšina mení stav, ktorý sa pamätá až do najbližšej zmeny
- vkladajú sa medzi operátory vstupu a výstupu
- `endl`, `flush`, `left/right`, `scientific`, `fixed`,  
`dec/oct/hex`, `skipws/noskipws`, `setprecision(int)`,  
`setw(int)`, `setfill(int)`
- `demo10_1.cc`

# Stav prúdu

- stream error flags
- *ios\_base::iostate*
  - *ios\_base* ako základná trieda udržujúca informácie o prúdoch
- goodbit (ok), eofbit (koniec súboru), failbit (posledná operácia nevyšla), badbit (pokus o neplatnú operáciu), hardbit (HW chyba)

# Stav prúdu

- metódy pre prístup
  - `ios::rdstate() const`
    - získanie error flags
  - `void ios::setstate(iostate stav)`
    - nastavenie príznaku
  - `bool ios::good() const`
    - je stream pripravený na IO operáciu?
  - `bool ios::eof() const;`
    - eofbit
  - `bool ios::fail() const;`
    - failbit/badbit
  - `bool ios::bad() const;`
    - badbit
  - `void ios::clear(iostate stav=goodbit)`
    - ak je prúd v chybovom stave a chceme ho použiť, musíme vymazať príznak

# Prúdy pre štandardné zariadenia

- `<iostream>`
- triedy
  - `istream`
    - vstupné operácie (`>>...`)
    - inštancie: `cin`
  - `ostream`
    - výstupné operácie (`<<...`)
    - inštancie: `cout`, `cerr`, `clog`
  - `iostream`
    - vstupno-výstupné operácie
- `demo10_2.cc`
  - ukážka použitia `istream`, `ostream`, preťaženia operátorov, zisťovanie stavu prúdu



# Súborové prúdy

- <fstream>
- triedy
  - ifstream
    - vstupný
  - ofstream
    - výstupný
  - fstream
    - vstupno-výstupný
- pri otváraní pomocou *fstream* špecifikujem mód (I,O,IO)
- demo10\_3.cc

# Súborové prúdy

- módy (flags):
  - *ios\_base*
  - kvalifikované *ios::*
  - použiteľné pre *fstream*, *ifstream*, *ofstream*
  - príklady:
    - *in* – otvorenie na čítanie
    - *out* – otvorenie na zápis
    - *ate* – otvorí súbor na konci, pozíciu môžem zmeniť cez *seekp()*
    - *app* – otvorí súbor na konci, pozíciu nezmením
    - *trunc* – ak súbor existuje, zmaže sa
    - *binary* - otvorenie binárneho súboru (bez *>>*, *<<*)
    - *nocreate*, *noreplace* – deprecated, nepoužívať

# Súborové prúdy

- upozornenie
  - meno súboru nie je *string*! (ale *char\**)
- *ios\_base::seek\_dir*
  - *beg* (posun voči začiatku), *cur* (voči aktuálnej pozícii), *end* (voči koncu súboru)
- *demo10\_4.cc*
  - načítanie súboru do pamäte

# Pamäťové prúdy

- `<sstream>`
- triedy
  - `istringstream`
    - vstupný
  - `ostringstream`
    - výstupný
  - `stringstream`
    - vstupno-výstupný
- využitie
  - konverzie, buffery, formátovanie v pamäti...

# Pamäťové prúdy

- neotvárajú sa, inicializujú sa konštruktorom  
ostreamstream prud("hello");
- metóda *str()*
  - *string str()* - vráti obsah
  - *void str(string s)* – nastaví obsah
- máme *sprintf()* a *sscanf()*
  - pracujú ako *printf()/scanf()* v C, ale nad pamäťou
- demo10\_5.cc

# Úloha 10 – Textové ovládajúce prvky

- vytvorenie tried reprezentujúcich sadu ovládacích prvkov
- nie úplne jednoduché na zorientovanie
- dostupné súbory, ktoré treba použiť
  - control.h, mycontrols.h, bitmap.h, bitmap.cc
  - odovzdávate mycontrols.h, mycontrols.cc
- prvky určené vzhľadom k rodičovi (kontrola)
- zadanie

[http://www.fi.muni.cz/usr/jkucera/pb161/text\\_control.htm](http://www.fi.muni.cz/usr/jkucera/pb161/text_control.htm)

+ komentáre v súboroch

# Dotazy?