

8. Text Parsing

Ján Dugáček

November 14, 2018

Table of Contents

- 1 Motivation
- 2 Parsing functions
 - `std::getline`
 - `std::stringstream`
 - `std::string`
 - Exercise
- 3 Homework

Motivation

- What if the file you need to read can't be read by `easy::vector`?

std::getline

```
easy::vector<std::string> lines(1);  
while (std::getline(in, lines.back(), '\\n')) {  
    if (!lines.back().empty())  
        lines.push_back();  
}
```

- std::getline reads from the input in its first argument (it can be std::cin, std::ifstream, ...), saves it into the string on its second argument until it finds the separator in its last parameter (new line by default)
- This program splits the input into lines, ignoring the empty ones
- Can you spot the error in the code?

Function #2

```
#include <sstream>
// ...
easy::vector<std::string> blocks;
std::stringstream stream(lines[i]);
std::string reading;
while (std::getline(stream, reading, '\\t')) {
    blocks.push_back();
    std::swap(blocks.back(), reading);
}
```

- `stringstream` can be used to parse a string in the same way as `std::cin` or `std::ifstream`

std::string

```
std::string text = "ahoy";  
text.push_back('!');  
char firstLetter = text[0];  
if (firstLetter >= 'A' && firstLetter <= 'Z')  
    firstLetter += 'A' - 'a';  
text[0] = firstLetter;  
if (text == "Ahoy!")  
    text.append(" Pleased to see you!");
```

- std::string can do plenty of stuff, but it may be impractical
- Mistakes or unhandled invalid inputs can cause crashes

std::string #2

```
int number = 4;
std::string text = "It's " + std::to_string(number);
text += "!";
int length = text.size();
for (unsigned int i = 0; text[i] != 0; i++) {
    if (text[i] >= '0' && text[i] <= '9')
        length--;
}
int number2 = std::stoi(std::to_string(number));
std::string theNumber = text.substr(5, 2);
```

- `std::to_string` can be used to convert a number to string
- `std::stoi` can be used to convert a string to an integer (`std::stof` for float)
- The final symbol in a string is always 0
- The `substr` method (in this case) returns a string containing 2 characters starting with the fifth

Exercise

- 1 Write a function that finds the number in string like The number is 14 probably.
- 2 Write a function that splits a comma-separated list string (like toaster,kettle,dishwasher)
- 3 Write a function that checks if a string is a number
- 4 Write a function that splits a string into capital letters, small letters and numbers and dumps the rest
- 5 Write a function that multiplies all numbers placed inside a string by a number given in the second argument

Homework

- Write a function that parses files as shown into a vector of vectors that are entries containing the x, y and z parametres respectively

x=1 y=2 z=3

y=1.1 x=8.2 z=0.3

z=0.1 y=12 x=6.1

- You have two weeks to do it