Отчёт по лабораторной работе №4 по теме «Декомпозиция программ»

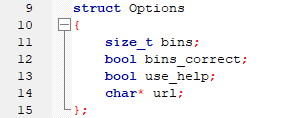
по предмету

Разработка Программного Обеспечения Систем Управления

Изотова Артёма Александровича, А-03-19

В моём варианте мне необходимо было сделать, так, чтобы bins вводилось через cmd консоль в качестве опции, причём опция может стоять как до, так и после url, если после bins числа не стоит нужно завершать программу и выводить подсказку.

Для реализации я ввёл новую структуру Options, которая содержит в себе непосредственно корзины bins, 2 логический переменные и указатель на url.



struct Options

{

size\_t bins;

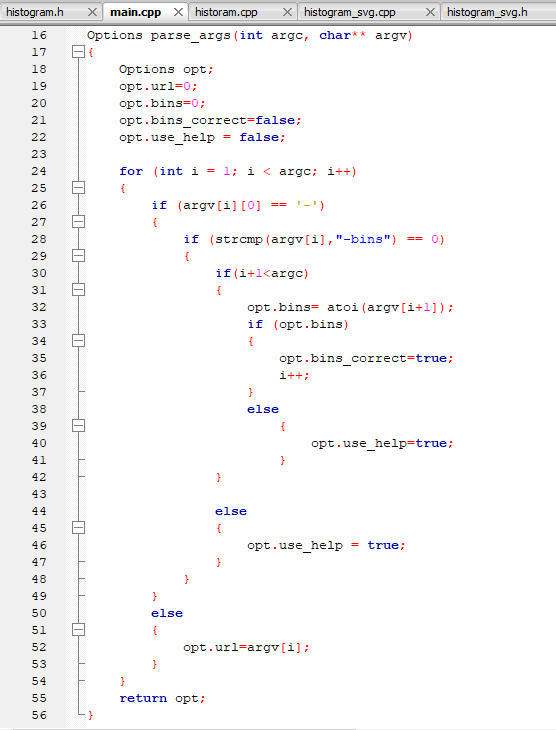
bool bins\_correct;

bool use\_help;

char\* url;

};

Также я создал новую функцию parse\_args, которая возвращает структуру. В этой функции я прохожу по символам, которые мы ввели в cmd, как только встречаю «-», то проверяю является ли это словом bins, после, если это bins, то я смотрю является ли следующий символ цифрой, для этого я присваиваю количеству корзин следующий символ, причём я также привожу его к типу integer, при помощи функции atoi, которая в случае, если это не цифра вернёт мне 0, следующим шагом я проверяю если там действительно цифра, то я я даю значение true для, того, чтобы понять, что у нас действительно есть цифра. И уже в main непосредственно проверяю, что если true, то в корзины спокойно записалась число и можем работать дальше, а если нет, то main возвращает 1(чтобы мы понял что ошибка) и в поток ошибок выводится ERROR.



Options parse\_args(int argc, char\*\* argv)

{

Options opt;

opt.url=0;

opt.bins=0;

opt.bins\_correct=false;

opt.use\_help = false;

for (int i = 1; i < argc; i++)

{

if (argv[i][0] == '-')

{

if (strcmp(argv[i],"-bins") == 0)

{

if(i+1<argc)

{

opt.bins= atoi(argv[i+1]);

if (opt.bins)

{

opt.bins\_correct=true;

i++;

}

else

{

opt.use\_help=true;

}

}

else

{

opt.use\_help = true;

}

}

}

else

{

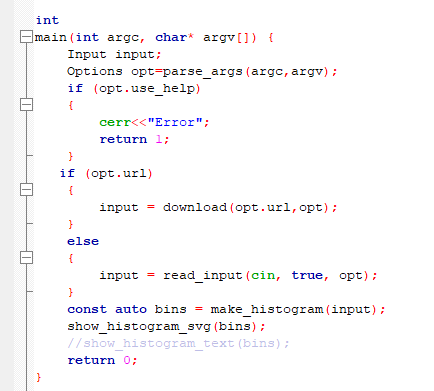
opt.url=argv[i];

}

}

return opt;

}



int

main(int argc, char\* argv[]) {

Input input;

Options opt=parse\_args(argc,argv);

if (opt.use\_help)

{

cerr<<"Error";

return 1;

}

if (opt.url)

{

input = download(opt.url,opt);

}

else

{

input = read\_input(cin, true, opt);

}

const auto bins = make\_histogram(input);

show\_histogram\_svg(bins);

//show\_histogram\_text(bins);

return 0;

}