

Технологично училище „Електронни системи“
Технически университет – София

Операционни системи

Задача 1: Изпълнение на проста аритметична функция на два операнда записана в текстов файл

Николай Алексиев

nbalexiev@elsys-bg.org



1. Условие на задачата

1.1 Основна функционалност (30 точки)

Целта на задачата е да се реализира програма, която прочита съдържанието на текстов файл (in.txt) и в случай, че то отговаря на условията за аритметичен израз поставени по-долу, извършва оказаното аритметично действие и записва резултата в двоичен файл (out.dat).

Аритметичният израз трябва да отговаря на следните условия:

- записан е на един ред;
- състои се от два целочислени операнда разделени от знак за аритметична операция;
- позволените аритметични операции са: „+“ и „-“;
- позволено е наличието на неограничен брой празни символи (интервал - ' ' и табулация - '\t') преди и след знака за аритметична операция;
- позволено е наличието на неограничен брой празни символи в началото и края на аритметичния израз.

Пример:

in.txt: “ 158 + 7 ”

out.txt: Съдържа двоично записано числото 165

1.2 Обработка на грешки (20 точки)

При липса на входен файл се извежда съобщение за грешка: „Input file not found!“

При неправилен формат на съдържанието записано във файла се извежда съобщение за грешка: „Illegal input format!“

2. Изисквания към решението

- 1) Програмата трябва да бъде написана на езика C съгласно ISO/IEC 9899:1990 (ANSI C).
- 2) Задължително е програмата да бъде разработена като проект под Eclipse CDT. Решението трябва да бъде изпратено под формата на tar.gz архив, който да включва целия проект.
- 3) При проверка на решението програмата ви ще бъде компилирани и тествана след импортиране на проекта в Eclipse CDT.
- 4) Работи, които са предадени по-късно от обявеното (или не са предадени), ще бъдат оценени с 0 точки.
- 5) За работа с файлове трябва да се използва семейството от функции за работа с файлови потоци FILE – fopen(), fclose(), fread(), fwrite(), fgetc(), fputc(), getline(), fgets(), fputs(), fread(), fwrite() и т.н. Неспазването на това изискване води до намаляване на оценката с 30%.
- 6) Правилата за оценяване са следните. Приемаме, че напълно коректна и написана спрямо изискванията програма получава максималния брой точки — 100%. Ако в решението има пропуски, максималният брой точки ще бъде намален съгласно правилата описани по-долу.

7) Програмата ви трябва да съдържа достатъчно коментари. Оценката на решения без коментари или с недостатъчно и/или мъгляви коментари ще бъде намалена с 30%.

8) Всеки файл от решението трябва да започва със следният коментар:

```
//-----  
// NAME: Ivan Ivanov  
// CLASS: Xia  
// NUMBER: 13  
// PROBLEM: #1  
// FILE NAME: xxxxxx.yyy.zzz (unix file name)  
// FILE PURPOSE:  
// няколко реда, които описват накратко  
// предназначението на файла  
//-----
```

9) Всяка функция във вашата програма трябва да включва кратко описание в следния формат:

```
//-----  
// FUNCTION: ххууzz (име на функцията)  
// предназначение на функцията  
// PARAMETERS:  
// списък с параметрите на функцията  
// и тяхното значение  
// FUNCTIONS CALLED:  
// списък на функциите, които се извикват  
// от описваната функция  
//-----
```

10) Лош стил на програмиране и липсващи заглавни коментари ще ви костват 30%.

11) Програми, които не се компилират получават 0 точки. Под „не се компилират“ се има предвид произволна причина, която може да причини неуспешна компилация, включително липсващи файлове, неправилни имена на файлове, синтактични грешки, неправилен или липсващ Makefile, и т.н. Обърнете внимание, че в UNIX имената на файловете са case sensitive.

12) Програми, които се компилират, но не работят, не могат да получат повече от 50%. Под „компилира се, но не работи“ се има предвид, че вие сте се опитали да решите проблема до известна степен, но не сте успели да направите пълно решение. Безсмислени или мъгляви програми ще бъдат оценявани с 0 точки, независимо че се компилират. Често срещан проблем, който спада към този случай, е че вашият Makefile генерира изпълним файл, но той е именуван с име, различно от очакваното (т.е. prog1 в разглеждания случай).

13) Програми, които дават неправилни или непълни резултати, или програми, в които изходът и/или форматирането се различава от изискванията ще получат не повече от 70%.

14) Всички наказателни точки се сумират. Например, ако вашата програма няма задължителните коментари в началото на файлове и функциите се отнемат 30%, ако няма достатъчно коментари се отнемат още 30%, компилира се, но не работи правилно — още 30%, то тогава резултатът ще бъде: $50 \cdot (100 - 30 - 30 - 30) \% = 50 \cdot 10 \% = 5$ точки.

15) Работете самостоятелно. Групи от работи, които имат твърде много прилики една с

друга, ще бъдат оценявани с 0 точки.

16)Вашата програма трябва да съдържа следните файлове:

- Файл `srtingArithmetic.c`, който съдържа решението на задачата.
- Файл `Makefile`, който компилира описаните по-горе файлове и създава изпълним файл. Обърнете внимание, че ако вашето решение не съдържа `Makefile` или съдържа некоректен `Makefile`, то най-вероятно ще попадне в категорията „не се компилира“. Допуска се използването на автоматично генериран `Makefile` от Eclipse CDT.