

ТЕХНОЛОГИЧНО УЧИЛИЩЕ “ЕЛЕКТРОННИ СИСТЕМИ”
ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ

ОПЕРАЦИОННИ СИСТЕМИ

Задача 4: Реализация на ls

`elsys.os.2013@gmail.com`



7 май 2014 г.

1 Условие на задачата

1.1 Основна функционалност (20 точки)

Целта на задачите е да се реализира стандартната UNIX команда `ls`.

Командата `ls` се използва за извеждане на съдържанието на директории. От командния ред програмата получава списък от директории и файлове, а на стандартния изход извежда имената на файловете, съдържащи се в всяка една от директориите.

Например:

```
$ ls /usr/share/texmf/tex
fontinst
generic
latex
```

Забележка: `ls` не показва скрити файлове.

В случай на списък от директории изходът изглежда по следния начин:

```
$ ls /usr/share/texmf/tex /usr/share/texmf/metapost/
/usr/share/texmf/metapost/:
metauml

/usr/share/texmf/tex:
fontinst
generic
latex
```

Когато на командата `ls` се предаде файл или списък с файлове, тя извежда имената на файловете:

```
$ ls a.txt
a.txt
$ ls a.txt b.txt
a.txt
b.txt
```

1.2 Поддръжка на `-a` (20 точки)

Командата `ls` трябва да поддържа незадължителният аргумент `-a`.

При използване на опцията `-a` командата трябва да изведе и скритите файлове. Например:

```
$ ls -a
total 20
drwxr-xr-x  2 hristo hristo 4096 May  7 22:06 .
drwxr-xr-x 75 hristo hristo 4096 May  7 22:05 ..
-rw-r--r--  1 hristo hristo   38 May  7 22:06 a.out
-rw-r--r--  1 hristo hristo   13 May  7 22:05 a.txt
-rw-r--r--  1 hristo hristo   20 May  7 22:05 b.txt
```

1.3 Поддръжка на `-l` (20 точки)

Командата `ls` трябва да поддържа незадължителният аргумент `-l`.

При използване на опцията `-l` командата трябва да изведе подробен информация за съдържанието на директорията. Например:

```
$ ls -l
total 1152
-rw-r--r-- 1 lubo wheel    18 2009-03-08 12:40 a.txt
-rw-r--r-- 1 lubo wheel    18 2009-03-08 12:41 b.txt
d----- 2 lubo wheel    48 2009-03-08 13:51 forbidden
```

1.4 Поддръжка на -R (20 точки)

Командата `ls` трябва да поддържа незадължителният аргумент `-R`.

При използване на опцията `-R` командата трябва рекурсивно да изведе информация за съдържанието на директорията. При примерна файлова структура:

```
.
|-- a.out
|-- a.txt
|-- b.txt
|-- dir1
|  |-- dir11
|  |-- file01.txt
|-- dir2
|-- dir3
```

Резултатът от изпълнението на програмата трябва да е:

```
$ ls -R

.:
a.out a.txt b.txt dir1 dir2 dir3

./dir1:
dir11 file01.txt

./dir1/dir11:

./dir2:

./dir3:
```

1.5 Комбинация от опции

Възможна е произволна комбинация от опции. Например: `ls -lRa`, `ls -al`, ..

1.6 Обработка на грешки (20 точки)

Ако някой от аргументите на `ls` не е файл или директория, то програмата трябва да изведе съобщение на стандартната грешка (`stderr`). Например, ако бъдат предадени аргументи, които не са файлове, то съобщението трябва да бъде оформено по следния начин:

```
$ ls aa
ls: cannot access aa: No such file or directory
```

Ако на програмата се предаде име на директория, която потребителят няма права да отвори, то съобщението трябва да бъде оформено по следния начин:

```
$ ls forbidden/
ls: cannot open directory forbidden/: Permission denied
```

2 Изисквания към решението и оценяване

1. Решението на задачата трябва да бъде написано на езика C съгласно ISO/IEC 9899:1999.
2. Правилата за оценяване са следните. Приемаме, че напълно коректна и написана спрямо изискванията програма получава максималния брой точки — 100% или 70 точки. Ако в решението има пропуски, максималният брой точки ще бъде намален съгласно правилата описани по-долу.
3. Задължително към файловете с решението трябва да е приложен и `Makefile`. Изпълнимият файл, който се създава по време на компилация на решението, трябва да се казва `ls`.

4. При проверка на решението програмата ви ще бъде компилирана и тествана по следния начин:

```
make
./ls .
```

Предходната процедура ще бъде изпълнена няколко пъти с различни входни данни за да се провери дали вашата програма работи коректно.

5. Реализацията на програмата трябва да спазва точно изискванията. Всяко отклонение от изискванията ще доведе до получаване на 0 точки за съответната част от условието.
6. Работи, които са предадени по-късно от обявеното (или не са предадени), ще бъдат оценени с 0 точки.
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
// ...
//-----
```

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
//-----
```

9. Лош стил на програмиране и липсващи заглавни коментари ще ви костват 30%.
10. 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”.
11. 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. 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. `ls` в разглеждания случай).
12. Безсмислени или мъгляви програми ще бъдат оценявани с 0 точки, независимо че се компилират.
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: $70 * (100 - 30 - 30 - 30)\% = 70 * 10\% = 7$ точки
15. Работете самостоятелно. Групи от работи, които имат твърде много прилики една с друга, ще бъдат оценявани с 0 точки.