Технологично училище "Електронни системи"
Технически университет – София

Операционни системи

Задача 1: Реализация на `tail`

elsys.os.2015@gmail.com

20 ноември 2015 г.

1 Условие на задачата

1.1 Основна функционалност (20 точки)

Целта на задачате е да се реализира стандартната `UNIX` команда `tail`.

Командата `tail` се използва за показване на последните редове на файлове. От командния ред програмата получава списък от файлове и започва да ги обработва като последните 10 реда от всеки файл последователно се копират в стандартния изход (`stdout`), разделени със заглавна секция.

Ако файлът съдържа по-малко от 10 реда, цялото негово съдържание се копира в стандартния изход.

Ако командата не получи аргументи от командния ред, тя започва да чете от стандартния вход.

Например, ако са дадени два файла със следното съдържание `a.txt`:

```
one
two
three
four
five
six
seven
eight
nine
ten
eleven
```

и `b.txt`:

```
1234 1234
567
890
```

то

```
$ tail a.txt b.txt
==> a.txt <==
two
three
four
five
six
seven
eight
nine
ten
eleven

==> b.txt <==
1234 1234
567
890
```

При наличието на само един аргумент, заглавната секция се пропуска.

1.2 Обработка на грешки (10 точки)

Ако някой от аргументите на `tail` не е файл или файлът не може да се отвори, то програмата трябва да изведе съобщение на стандартната грешка (`stderr`). Например, ако бъдат предадени аргументи, които не са файлове

```
tail aa bb
```

съобщението трябва да бъде оформено по следния начин:

```
tail: cannot open 'aa' for reading: No such file or directory
tail: cannot open 'bb' for reading: No such file or directory
```

Ако на програмата се предаде име на файл, който потребителят няма права да отвори

```
tail /etc/shadow-
```

съобщението трябва да бъде оформено по следния начин:

```
tail: cannot open '/etc/shadow-' for reading: Permission denied
```

1.3 Поддръжка на стандартен вход (10 точки)

Когато командата получи от командния ред аргумент `-`, тя трябва да интерпретира този аргумент като стандартния вход. Например командата

```
tail f1.txt - f2.txt
```

трябва да изпълни следната последователност от действия:

1. Да копира последните 10 реда на файла `f1.txt` на стандартия изход.
2. Да чете редове от стандартния вход, докато не бъде затворен и да копира последните 10 реда на стандартния изход.
3. Да копира последните 10 реда на файла `f2.txt` на стандартия изход.

2 Изисквания към решението и оценяване

1. Програмата трябва да бъде написана 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 40 точки. Ако в решението има пропуски, максималният брой точки ще бъде намален съгласно правилата описани по-долу.
3. За работа с файлове трябва да се използва семейството от функции за работа с файлове от ниско ниво `open()`, `close()`, `read()`, `write()`, и т.н. Неспазването на това изискване води до оценяване с 0 точки.
4. 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 `tail`.
5. При проверка на решението програмата ви ще бъде компилирани и тествана по следния начин:

   ```
   make
   ./tail f1.txt
   ```

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

6. Реализацията на програмата трябва да спазва точно изискванията описани по-горе. Всяко отклонение от изискванията ще доведе до получаване на 0 точки за съответната част от условието.

7. Работи, които са предадени по-късно от обявеното (или не са предадени), ще бъдат оценени с 0 точки.

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

9. 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: xxyyzz (име на функцията)
//  предназначение на функцията
// PARAMETERS:
//  списък с параметрите на функцията
//  и тяхното значение
//--------------------------------------------
```

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. 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. `tail` в разглеждания случай).

13. Безсмислени или мъгляви програми ще бъдат оценявани с 0 точки, независимо че се компилират.

14. 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%.

15. 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: $50 * (100 - 30 - 30 - 30)\% = 50 * 10\% = 5$ точки

16. Работете самостоятелно. Групи от работи, които имат твърде много прилики една с друга, ще бъдат оценявани с 0 точки.