Упражнение 1 - задачи

- Да се напише програма на C, която реализира командата **cp файл1 файл2**
- Да се напише програма на C, която реализира командата **cat** (без параметри)
- Да се напише програма на C, която реализира командата **cat >>файл**
- Да се напише програма на C, която реализира командата

  tee [файл1 файл2 ..... файлN]

- Да се напише програма на C, която реализира командата

  wc [-cwl] [файл1 файл2 ....файлN]

- Да се напише програма на C, която реализира командата **head -10l файл**
- Да се напише програма на C, която реализира командата **cut -cm-n** файл
  (m и n да се четат от входа)

Упражнение 2

1. Създайте файл **aa** със съдържание английската азбука извън програмата. Задаяата отваря файла **aa** само за четене. Четете от него с цикъл **for** и променлива променяща се от 5 до 1 в буфер с размер 16 байта. Отпечатайте с **printf** прочетеното в буфера на всяка итерация.

2. Горната задача, като вместо **printf** използвайте системен примитив **write** и изведете на стандартния изход.

3. Горните 2 задачи като промените съдържанието на файла **aa** на следното – abcdefghij\n.

4. Увеличете цикъла от 8 до 1 в горните задачи.

5. Първата задача като преди всяко **read** слагате системен примитив **lseek**.
    1) lseek(fd, 0 , 0)
    2) lseek(fd, 3, 0)
    3) lseek(fd, 3, SEEK_CUR)
    4) lseek(fd, -3, SEEK_END) и други варианти

6. Файлът се подава като параметър в командния ред.

7. В цикъла изпълняваме команда, която брои буквите във файла **aa.**

8. Запишете във файл 25 байта извън програмата. Програмата чете от файла по 10 байта и отпечатава какво връща състемния прмитив **read**

9. Някоя от общите задачи от миналия път.

Упражнение 3

1. Напишете програма, която създава файла **fff** и го отваря три пъти:
   - само за четене през fd1
   - само за писане през fd2
   - за четене и писане през fd3

   След това:

   А)
   - през fd2 записва “Hello world” в него
   - през fd3 прочита 6 байта и изписва прочетеното на стандартния изход
   - през fd1 прочита 5 байта и изписва прочетеното на стандартния изход
   - през fd3 записва “chil” в него
   - през fd2 записва “!!!” в него
   - през fd1 прочита 9 байта и изписва прочетеното на стандартния изход

   Б)
   - през fd3 записва “Hello” в него
   - през fd2 записва „worlds” в него
   - през fd1 прочита 6 байта и изписва прочетеното на стандартния изход
   - затваря fd2
   - през fd3 записва “oops” в него
   - през fd1 прочита 6 байта и изписва прочетеното на стандартния изход

   Какво ще се изпише на стандартния изход и какво ще бъде съдържанието на файла във всеки от горните случаи?

2. В текущата директория се намира файл **f1** със съдържание:
   abcd-abcd-abcd
   Отворете файла с 2 различни дескриптора за четене и писане. Започвайки от последните 2 символа напишете думата **wxyz**. Запишете върху 5 и 6-тия символ от началото на файла - цифрите 12. Отпечатайте цялото съдържание на файла.

3. Да се напише програма на Си, която получава като параметри в командния ред число и име на файл. Програмата извежда на стандартиния изход съдължанието на файла от указания от числото ред.

4. Трасировки:
    - Какво ще бъде изведено на стандартиня изход след успешното изпълнение на програмата /home/KN/SYSPROG/LAST/tras_1.
    - Файлът ffff е със съдържание **I am a child**! Какво ще бъде изведено на стандартния изход и какво ще е съдържанието на файла ffff след успешното изпълнение на програмата /home/KN/SYSPROG/LAST/tras_2.

5. Да се състави програма на Си, която получава два параметъра, първият от които е име на съществуващ текстов файл, в който дължината на редовете не надвишава 60. Програмата извежда на стандартния изход онези от редове от текстовия файл, чийто последен символ е $. Създава файл с име втория параметър и в него записва - всеки път на нов ред – само цифрите, от всеки от останалите редове от първия файл. Извежда броя на редовете от новосформирания файл.

Упражнение 4

1. Напишете програма на Си, която отваря файл за четене, дублира дескритора, последователно чете от двата дескриптора и извежда прочетеното на екрана. Затваря оригиналния дексриптор и продължава за чете от дублирания.
2. Напишете програма на Си, която печата символния низ „EXAMPLE“ 10 пъти на стандартния изход.

3. В задача 2 се създава файл с име f1 и се пранасочва стандартния изход в него :
    - чрез open
    - чрез системен примитив dup

4. Името на файла се подава като параметър на програмата

5. Напишете програма на Си, която печата симолния низ „EXAMPLE“ и символния низ „HELLO“ 10 пъти на стандартния изход, като между тях се извеждат числата от 1 до 10.

6. В зад 5 след EXAMPLE се пранасочва стандартния изход в подадения като параметър файл и числата от 1 до 10 излизат във файла. След това се възстановява стандартния изход и HELLO отново е на екрана.

7. Напишете програма на Си, която получава като аргумент два файла. Чете от първия файл и извежда във втория чрез механизма на пренасочване на стандартния вход и изход.

8. Примери за пренасочване на стандартния вход и стандартния изход за грешки.

9. Трасировки:
    - /home/KN/SYSPROG/LAST/upr4_1
    - /home/KN/SYSPROG/LAST/upr4_2 с файл fileA

Упражнение 5 - задачи

Задача 1. Да се напише програма на С, която използвайки системни примитиви за работа с файлове, чете последователност от символи от стандартния вход. Добавя ги след края на файл, чието име е подадено като първи параметър / ако файла не съществува се създава/. Замества символите за табулация с ‘ | -> ’ и резултата записва във файл, чието име е подадено като втори параметър / ако файл съществува – старото съдържание се изтрива/.

Задача 2 Да се напише програма на С, която използвайки системни примитиви за работа с файлове, чете последователно по един символ от файл, чието име е подено като първи параметър и по един символ от стандартния вход. Добавя прочетените символи, премахвайки символите ‘0’, ’1’, ’2’, ’3’, ’4’, ’5’, ’6’, ’7, ’8, ’9’, последователно след края на файл, чието име е подадено като втори параметър/ ако файла несъществува се създава/. Чете се до достъгане на края на файла или до изчерпване на входния поток.

Задача 3. В текущия каталог се намира текстов файл **ABC** със следното съдържание:

Let’s go light up the world!

Напишете какво ще има записано във файла и какво ще бъде изведено на стандартния изход като резултат от изпълнението на дадения по-долу фрагмент от програма на С :

```
main( )
{
int fdi, fdio, nb1, nb2;
char buff [ 15 ] ;
        fdi = open ( “ ABC”, O_RDONLY);
         fdio = open ( “ ABC”, O_RDWR);

         nb1= read( fdi,buf,6);
         write ( 1, buf, nb1);

        lseek( fdi,3,SEEK_CUR);
        nb1= read( fdi,buf,6);
        write ( 1, buf, nb1);

        lseek( fdio,-20,SEEK_END);
```

```
        write( fdio,"disc",4);

         write( fdio,"over",4);
         nb2=read( fdio,buf,5);
         write ( 1, buf, nb2);
         write ( 1,"sky", 3);

         lseek( fdi,-2,SEEK_END);
         nb2= read( fdi,buf,2);
         write ( 1, buf, nb2);
}
```

Задача 4. В текущия каталог се намира текстов файл **ABC.** Като първи параметър на програма на C се подава съществуващ празен файл. Напишете какво ще има записано във файла и какво ще бъде изведено на стандартния изход като резултат от изпълнението на дадения по-долу фрагмент от програма на C :

```
#include      <unistd.h>
#include      <fcntl.h>
#include      <stdio.h>
main ( int argc, char *argv[] )
{
        int fd, i, status;
          if ( fork() )
         { wait(&status);
           for ( i=0; i<=4; i++)
          { write ( 1, "cat\n",4);
            execlp ( "cat", "cat", "ABC",0);
            write ( 1, "test1\n",5); }
           }
          else { close(1);
                 fd = open ( argv[1], O_RDWR) ;
              write (1, "test2\n",5);}
write (1, "test3\n",5);
}
```

Задача 5. В текущия каталог се намира текстов файл **abc** със следното съдържание :

Let's go change the world!

Напишете какво ще има записано във файла и какво ще бъде изведено на стандартния изход като резултат от изпълнението на дадения по-долу фрагмент от програма на C :

```
#include      <fcntl.h>
#include      <stdio.h>
main( )
{
int fdi, fdio, nb1, nb2;
char buff [ 15 ] ;
        fdi = open ( " abc", O_RDONLY);
         fdio = open ( " abc", O_RDWR);

         nb1= read( fdi,buf,6);
         write ( 1, buf, nb1);

        lseek( fdi,3,SEEK_CUR);
        nb1= read( fdi,buf,6);
        write ( 1, buf, nb1);

        lseek( fdio,-18,SEEK_END);
       write( fdio,"tra",3);

        write( fdio,"vel",3);
        nb2=read( fdio,buf,5);
        write ( 1, buf, nb2);
        write ( 1,"time", 4);

        lseek( fdi,-2,SEEK_END);
        nb2= read( fdi,buf,2);
        write ( 1, buf, nb2);
}
```

Упражнение 6

1. Напишете програма на Си, която изпълнява команда извеждаща редовете, в които се среща символния низ int в подадения като параметър файл.
2. В зад 1 – напишете грешна команда.
3. Да се напише програма на С, която получава като параметър команда (без параметри) и при успешното ѝ изпълнение, извежда на стандартния изход името на командата.
4. Да се напише програма на С, която получава като параметър команда с опции и аргументи. При успешното ѝ изпълнение, извежда на стандартния изход кода на завършане.
5. Да се напише програма на С, която получава като параметри три команди (без параметри), изпълнява ги последователно, като изчаква края на всяка и извежда на стандартния изход номера на завършилия процес, както и неговия код на завършване
6. Да се напише програма на С, която получава като параметър име на файл. Отваря подадения файл за писане (ако не съществува, го създава, в противен случай го занулява), създава 2 процеса и двата процеса пишат символния низ Hello във файла.

    - чрез единствения дескриптор
    - файла се отваря с 2 различни дескриптора.
    - файла се отваря с втори дескриптор, дублиращ първия

7. Да се напише програма на С, която получава като параметър име на файл. Създава процес син, който записва стринга Hello във файла (ако не съществува, го създава, в противен случай го занулява), след което процеса родител прочита записаното във файла съдържание и го извежда на стандартния изход, добавяйки по един интервал между всеки два символа.
8. Да се напише програма на С, която получава като параметри имена на два файла. В зад 6 пренасочва изхода на процеса дете във втория файл (ако не съществува, го създава, в противен случай добавя в него)
9. Трасировка – upr6_tras1.c; upr6_tras2.c; upr6_tras3.c; upr6_tras4.c

Упражнение 7

1. Да се пуснат един след друг два системни примитива fork(), като за всеки генериран процес се отпечатва pid-а на процеса/getpid()/ и pid-а на родителския процес/getppid()/.
2. Да се пусне един fork() и само в детския процес още един fork(). За всеки генериран процес се отпечатва pid-а на процеса/getpid()/ и pid-а на родителския процес/getppid()/.
3. Да се пуснат три fork() един след друг. Колко процеса се генерират?

4. Да се напише програма на C, която получава като параметри две имена на файлове. Отваря първия файл за писане (ако не съществува, го създава, в противен случай го занулява), създава 2 процеса. Родителят пише hello1, а детето – hello2 във първия файл.
5. В зад 4 пренасочвате изхода на детето във втория файл. В общи действия на двата процеса се записват hello3 на стандартния изход и hello4 във втория файл:
    - с wait в родителя
    - без wait в родителя
6. В зад 5 в края на локалните действия на детето възстановявате стандартния му изход.
7. В зад 6 – да се пробва с exit() – в единия или двата процеса; с или без wait()

8. Да се напише програма на C, която получава като параметри от командния ред две команди (без параметри). Изпълнява първата. Ако тя е завършила успешно изпълнява втората. Ако не, завършва с код -1.
9. Да се напише програма на C, която получава като командни параметри две команди (без параметри). Изпълнява ги едновременно и извежда на стандартния изход номера на процеса на първата завършила успешно. Ако нито една не завърши успешно извежда -1

10. Трасировка – upr7_tras1; upr7_tras2; upr7_tras3

Упражнение 8

Описание на примерните стъпки, включени в реализацията на прост вариант на програма команден интерпретатор /КИ/ - изпълнение на команди без опции и аргументи

Основни стъпки :

- цикъл, в който :
  - се извежда промпт
  - чете се в низ символ по символ ред от файла на стандартния вход
    като
    - интервалите се пропускат
    - (тук впоследствие се включват проверките и подготовка за обработване на специалните за КИ символи >, <, &, |)
    - символа '\n' се заменя с '\0'
    - третира се фонов режим/&/

    прочитане на конкретна команда (ключова дума) - *logout / quit / bye* и пр.
    /
    до или
    \
    достигане края на файла на стандартния вход
  - изпълнява се примитив fork, при който
    - стартира детски процес, в рамките на който се стартира команда с име прочетения от входа низ
    - родителският процес изчаква завършването на детския

По желание:

Да се напише програма на C, която получава като параметри от команден ред две команди (без параметри) и име на файл в текущата директория. Ако файлът не съществува, го създава. Програмата изпълнява командите последователно, по реда на подаването им. Ако първата команда завърши успешно, програмата добавя нейното име към съдържанието на файла, подаден като команден параметър. Ако командата завърши неуспешно, програмата уведомява потребителя чрез подходящо съобщение.