# Упражнение 15 - трасировка

**Задача 1**. Какво ще бъде изведено на стандартния изход при успешно изпълнение на дадения по-долу фрагмент от програма на Си (заградете верния отговор)

………

int j = 7 ;

if ( fork() = = 0 ) {

j++ ; exit(0);

if ( execlp (“who”,”who”,0 ) = = -1 ) j = 7;

else printf (“\nStoinostta na j = %d “, j);

}

else { --j ; printf (“\nStoinostta na j = %d “, j ) ; }

а) Stoinostta na j = 8 ;

Stoinostta na j = 6;

б) Stoinostta na j = 6;

в) Stoinostta na j = 7;

г) изходът от изпълнението на командата who

д) нито един от посочените случаи

**Задача 2** Като първи параметър на програма на С 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 С :

#include ……………..

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

{ int fd, k, status;

fd = open ( argv[1], O\_RDWR) ;

if ( fork() )

{ wait(&status);

dup(fd);

write ( 1, “who\n”, 4);

execlp ( “who”, “who”,”-u”,0);

write (fd, “ exam1\n”,5);

}

else { close(1); dup(fd);

for ( k=0; k<=3; k++)

write (1, “exam2\n”,6);

}

execlp ( “pwd”, “pwd”,0);

}

**Задача 3** В текущия каталог се намира текстов файл fileD.txt със следното съдържание

123456789

$$abcdefg

Напишете вдясно какво ще бъде изведено на стандартния изход (терминала)и какво ще бъде съдържанието на новосъздадения фаил, като резултат от изпълнението на файла, получен при успешна компилация на зададения по-долу програмен код на С, в който са използвани системни примитиви на ОС UNIX и LINUX:

#include <stdio.h>

#include <fcntl.h>

main( )

{ int fdr, fdw, n\_byt, i = 0 , status;

char symbols [ 40 ], c ;

if ( fork( ) )

{ wait ( &status);

if ( open ("file\_new", O\_RDONLY ) != -1 )

{ write ( 1, "\n", 1 );

execlp (“wc", "wc", "-c", "file\_new",0 );

execlp (“head", "head", "-5c", "file\_new",0 );

}

}

else {

if ( ( fdr = open ("fileD.txt",O\_RDONLY ) ) = = -1 )

{ printf ("\n Cannot open \n" ); exit (1); }

if (( fdw=open ("file\_new", O\_CREAT|O\_TRUNC|O\_RDWR,0666 )) == -1 )

{ printf ("\n Cannot creat \n" ); exit (1); }

n\_byt = read ( fdr , symbols, 40 );

c = symbols[ i++];

if ( c >= '0' && c <= '9' )

{ while ( symbols [ i ++] != '\n' && i < n\_byt )

write ( fdw, "$", 1 );

write ( fdw, "\n", 1 );

write ( 1, "\n", 1 );

}

else { write( 1, symbols, n\_byt ); write ( 1, "\n", 1 );

}

write ( fdw, "$\n", 2 );

write ( 1, symbols, 12);

close ( fdr); close (fdw);

}

}