НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ

**Лабораторна робота №1, №9.** Базові інструментальні засоби Linux. Створення й завершення процесів.

Виконав:

Студент групи КН-920в

Єрмаков Матвій Володимирович

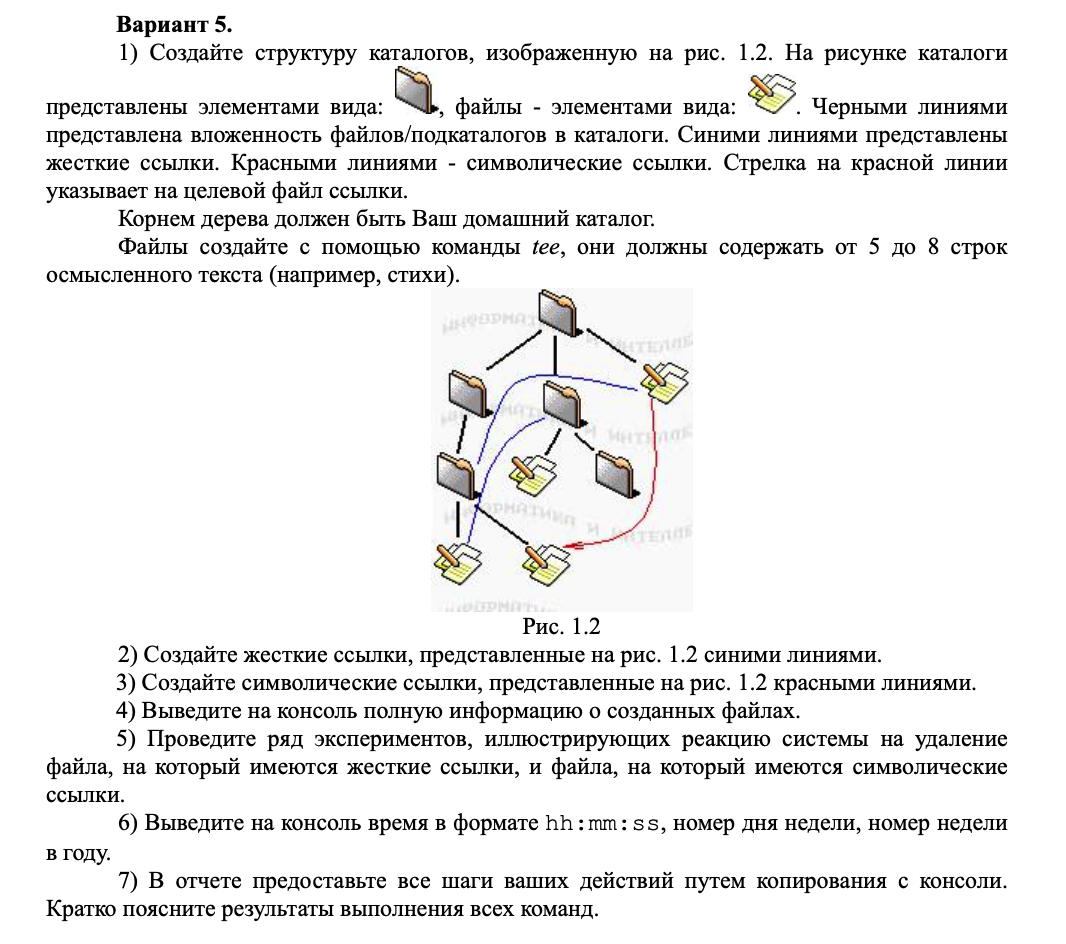
Харків 2022

**Хід роботи лабораторної роботи №1**

**Тема:** Базові інструментальні засоби Linux.

**Мета:** Ознайомлення з базовими інструментальними засобами командного інтерпретатора Linux.

**Індивідуальне завдання:**



*Рисунок 1 - Індивідуальне завдання.*

*Створюємо каталоги за допомогою команди mkdir в головній директорії bash:*  
  
matvejermakov@MacBook-Air-Matvej bash% mkdir 'cat\_1'

matvejermakov@MacBook-Air-Matvej bash % mkdir 'cat\_2'  
matvejermakov@MacBook-Air-Matvej bash % cd cat\_1

matvejermakov@MacBook-Air-Matvej cat\_1 % mkdir 'content'

matvejermakov@MacBook-Air-Matvej cat\_1 % cd content  
  
*Затим, створюємо 2 файли з віршами:*

matvejermakov@MacBook-Air-Matvej content % nano file1

matvejermakov@MacBook-Air-Matvej content % nano file2  
matvejermakov@MacBook-Air-Matvej content % cat file1

Украдкой время с тонким мастерством

Волшебный праздник создает для глаз.

И в то же время в беге круговом

Уносит все, что радовало нас.

Часов и дней безудержный поток

Уводит лето в сумрак зимних дней,

Где нет листвы, застыл в деревьях сок,

Земля мертва/и белый плащ на ней.

И только аромат цветущих роз -

Летучий пленник, запертый в стекле, -

Напоминает в стужу и мороз

О том, что лето было на земле.

Свой прежний блеск утратили цветы,

Но сохранили душу красоты.

matvejermakov@MacBook-Air-Matvej content % cat file2

Однажды старушка

У нас в городке

Послала на мельницу

Мышку в мешке.

Но мельник ни разу

Мышей не молол,

А если молол,

То не брал за помол.

*Виходимо в корінну папку:*

matvejermakov@MacBook-Air-Matvej content % cd ..

matvejermakov@MacBook-Air-Matvej cat\_1 % cd ..

matvejermakov@MacBook-Air-Matvej bash % cd cat\_2

matvejermakov@MacBook-Air-Matvej cat\_2 % mkdir content

matvejermakov@MacBook-Air-Matvej cat\_2 % nano file3

matvejermakov@MacBook-Air-Matvej cat\_2 % cd ..

*Створюємо символічне посилання “symbolicLink” на файл file2 в cat\_1/content:*

matvejermakov@MacBook-Air-Matvej bash % ln -s symbolicLink cat\_1/content/file2

*Виводимо вміст каталогів за допомогою команди tree:*

matvejermakov@MacBook-Air-Matvej bash % tree -a

.

├── .DS\_Store

├── cat\_1

│ ├── .DS\_Store

│ └── content

│ ├── file1

│ ├── file2

│ └── symbolicLink

├── cat\_2

│ ├── .DS\_Store

│ ├── content

│ ├── file1

│ └── file3

└── symbolicLink -> cat\_1/content/file2

4 directories, 9 files

*Видалимо основний файл “file1”, і затим запустимо файл жорсткого посилання на нього, який був створений у директорії cat\_2:*

matvejermakov@MacBook-Air-Matvej cat\_2 % cat file1

Украдкой время с тонким мастерством

Волшебный праздник создает для глаз.

И в то же время в беге круговом

Уносит все, что радовало нас.

Часов и дней безудержный поток

Уводит лето в сумрак зимних дней,

Где нет листвы, застыл в деревьях сок,

Земля мертва/и белый плащ на ней.

И только аромат цветущих роз -

Летучий пленник, запертый в стекле, -

Напоминает в стужу и мороз

О том, что лето было на земле.

Свой прежний блеск утратили цветы,

Но сохранили душу красоты.

*І бачимо, що все прекрасно працює.  
  
Видалимо file2 і й запустимо файл symbolicLink с символічний посиланням на цей же файл:*

*До видалення:*

matvejermakov@MacBook-Air-Matvej bash % cat symbolicLink

Однажды старушка

У нас в городке

Послала на мельницу

Мышку в мешке.

Но мельник ни разу

Мышей не молол,

А если молол,

То не брал за помол.

*Після:*   
matvejermakov@MacBook-Air-Matvej bash % cat symbolicLink

cat: symbolicLink: No such file or directory

*Результат каже на про те, що нема такого файлу.*

*Виводимо дату у форматі hh:mm:ss, номер дня тижня, номер тижня в році:*

matvejermakov@MacBook-Air-Matvej ~ % date +"%H:%M:%S, %u, %U"

17:55:02, 3, 07

**Висновок:** в ході даної лабораторної роботи я навчився користуватися терміналом Mac OS, а також навчився працювати з файли, задавати прямі та символічні посилання.

**Хід роботи лабораторної роботи №9**

**Тема:** Створення й завершення процесів.

**Мета:** Оволодіння системними програмними засобами породження та завершення процесів у ОС Linux.

**Індивідуальне завдання:**

Выполнить порождение 7 процессов-потомков. В каждом из выполняющихся процессов определить значение своего и родительского идентификатора. От каждого родителя выполнить передачу сообщения «Здравствуй, Сын! ID моего родителя =, Мой ID =, Твой ID= !» своему «сыну» через файл. Завершать каждый процесс только после завершения всех его потомков.

**Код програми:**

#include <stdio.h>

#include <stdlib.h>

#include <errno.h>

#include <unistd.h>

#include <string.h>

#include <signal.h>

#define PROCESS\_QUANTITY 9

void child(int i)

{

FILE \* t = fopen("file\_v4", "w");

fprintf(t, "Child number %d, ID %d Son: %d Parent %d \n", i, getpid(), getpid() + 1, getppid());

int pid;

if (i == PROCESS\_QUANTITY)

return;

pid = fork();

if (pid < 0){

;

}

else if (pid == 0)

child(++i);

else

waitpid(pid, NULL, 0);

fclose(t);

exit(0);

}

int main() {

int i = 0;

int pid = fork();

if (pid < 0)

;

else if (pid == 0)

child(++i);

else

waitpid(pid, NULL, 0);

char strMass[PROCESS\_QUANTITY + 1][80];

FILE \* u = fopen("file\_v4", "r");

int k = 0,

l = 0;

while(!feof(u)){

char j = fgetc(u);

if(j == '\n'){

k++;

l = 0;

}

else{

strMass[k][l] = j;

l++;

}

}

for(int i = PROCESS\_QUANTITY -2;

i >= 1;

--i)

printf("%s\n", strMass[i]);

fclose(u);

exit(1);

}

**Результат дії програми:**

matvejermakov@MacBook-Air-Matvej c % gcc -o main main.c -lm

matvejermakov@MacBook-Air-Matvej c % ./main

Child number 1, ID 43637 Son: 43638 Parent 43636

Child number 2, ID 43638 Son: 43639 Parent 43637

Child number 3, ID 43639 Son: 43640 Parent 43638

Child number 4, ID 43640 Son: 43641 Parent 43639

Child number 5, ID 43641 Son: 43642 Parent 43640

Child number 6, ID 43642 Son: 43643 Parent 43641

Child number 7, ID 43643 Son: 43644 Parent 43642

**Опис дії програми:**

Запускаємо процес за допомогою команди *fork()* та кладемо його у змінну. Тоді ми перевіряємо його на дочірність: якщо це так, то викликаємо метод *child(),* у якому методом ітерації,поки *і* не буде рівнятися заданій кількості дочірніх процесів, запускаємо новий дочірній процес і заносимо *id* його сина, його батька, й *id* його самого у файл *“file\_v4”.* Після завершення дії функції *child()* зчитуємо дані з файлу й виводимо результат на екран.

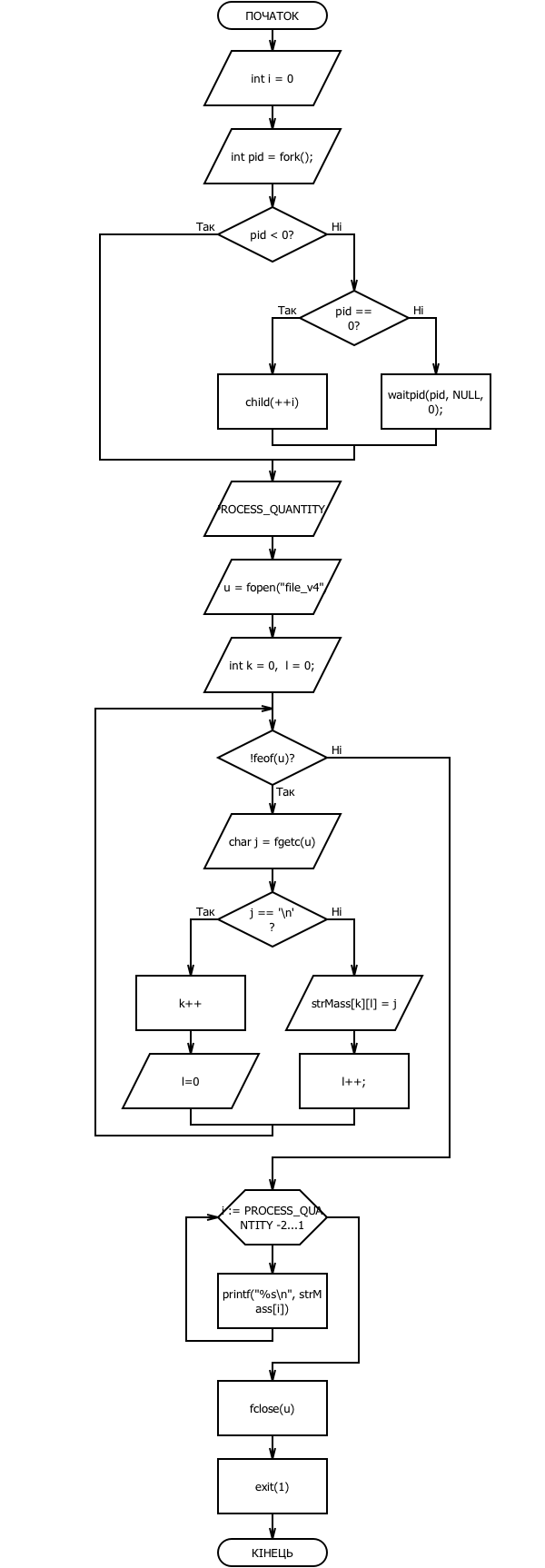
****

Рисунок 2. Блок-схема програми

**Висновок:** я ознайомився з базовими інструментальними засобами командного інтерпретатора Linux, навчився запускати батьківські й дочірні процеси.