

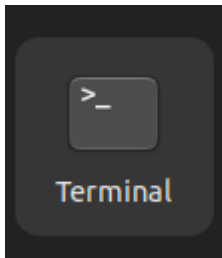
Sprawozdanie z laboratoriów komputerowych

Zad 1 sprawdzenie funkcjonalności poleceń

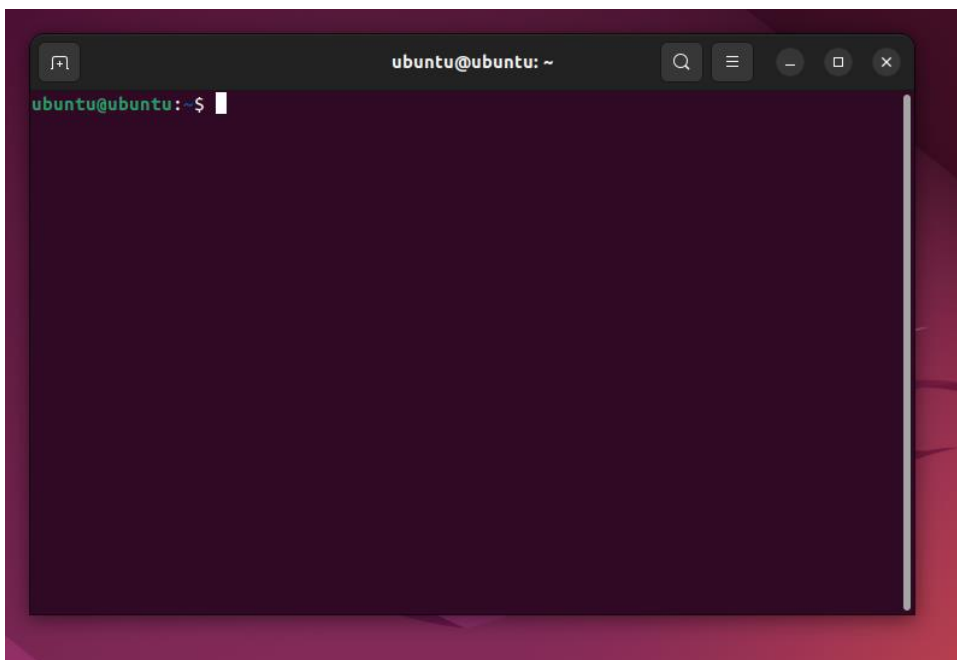
Uruchamiamy terminal wchodząc najpierw do wyboru programów:



Klikamy na terminal



Po uruchomieniu wyświetla nam się następujące okno:



Aby sprawdzić każde polecenie, wpisujemy je w terminal

Wpisujemy polecenie ls i otrzymujemy:

```
ubuntu@ubuntu:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
```

Widzimy, że komenda ls pokazuje nam wszystkie foldery i pliki znajdujące się w katalogu, w którym jest użytkownik terminalu.

Następnie wpisujemy polecenie mkdir. Jeżeli nie wpisujemy żadnych argumentów otrzymamy następującą odpowiedź:

```
ubuntu@ubuntu:~$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
```

Możemy wpisać argument `--help` aby zobaczyć co robi to polecenie. Po wpisaniu tego argumentu otrzymujemy następującą odpowiedź:

```
ubuntu@ubuntu:~$ mkdir --help
Usage: mkdir [OPTION]... DIRECTORY...
Create the DIRECTORY(ies), if they do not already exist.
```

Widzimy, że polecenie `mkdir` jest używane do tworzenia nowych folderów, możemy teraz wpisać ten sam argument dla wszystkich innych komend wymagających sprawdzenia

Dla polecenia `mv`:

```
ubuntu@ubuntu:~$ mv --help
Usage: mv [OPTION]... [-T] SOURCE DEST
  or: mv [OPTION]... SOURCE... DIRECTORY
  or: mv [OPTION]... -t DIRECTORY SOURCE...
Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.
```

Widzimy, że polecenie `mv` jest używane do przenoszenia folderów

Dla polecenia `rm`:

```
ubuntu@ubuntu:~$ rm --help
Usage: rm [OPTION]... [FILE]...
Remove (unlink) the FILE(s).
```

Widzimy, że polecenie `rm` używane jest do usuwania plików

Dla polecenia `cp`:

```
ubuntu@ubuntu:~$ cp --help
Usage: cp [OPTION]... [-T] SOURCE DEST
  or: cp [OPTION]... SOURCE... DIRECTORY
  or: cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.
```

Widzimy, że polecenie `cp` używane jest do kopiowania plików.

Zad 2

Aby utworzyć folder możemy użyć wcześniej opisanego polecenia `mkdir`. Wchodzimy najpierw do folderu w którym chcemy utworzyć folder:

```
ubuntu@ubuntu:~$ cd Desktop
```

Następnie wpisujemy polecenie z następującymi argumentami:

```
ubuntu@ubuntu:~/Desktop$ mkdir test
```

Po wpisaniu `ls` widzimy, że folder `test` został utworzony:

```
ubuntu@ubuntu:~/Desktop$ ls
test  ubiquity.desktop
```

Teraz aby usunąć folder test wystarczy że użyjemy polecenie rm nie zmieniając folderu w następujący sposób:

```
ubuntu@ubuntu:~/Desktop$ rm --dir test
```

Gdzie argument `--dir` oznacza, że polecenie może usuwać puste foldery.

Wpisując ls widzimy że folder test został usunięty:

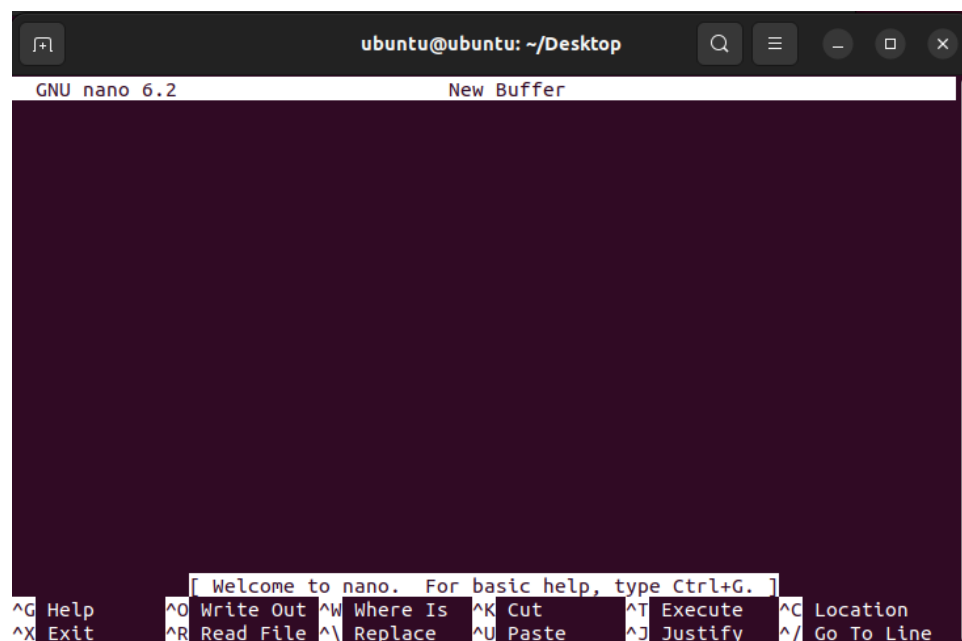
```
ubuntu@ubuntu:~/Desktop$ ls
ubiquity.desktop
```

Zad 3

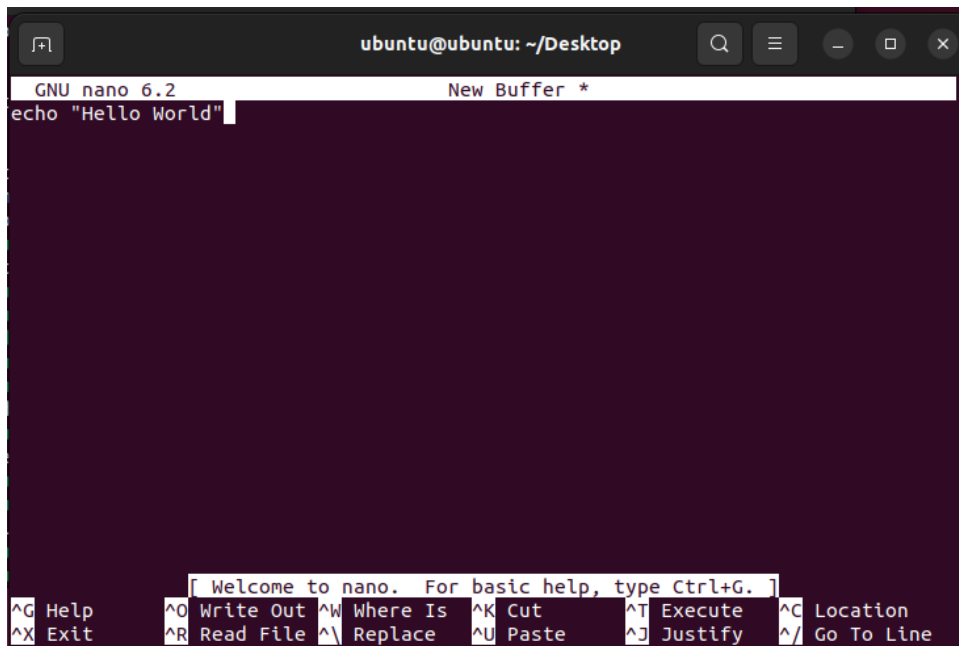
Aby utworzyć nowy skrypt bash wpisujemy polecenie nano do terminala:

```
ubuntu@ubuntu:~/Desktop$ nano
```

Wyświetla nam się następujące pole do edycji skryptów:

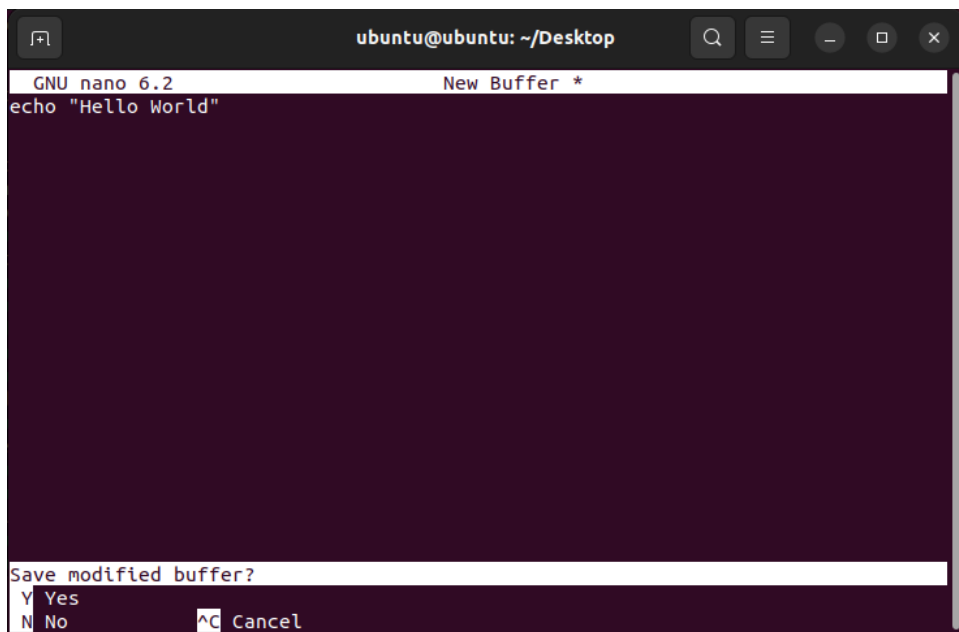


Aby skrypt wyświetlił nam napis „Hello world” wystarczy że wpisujemy następujący tekst:



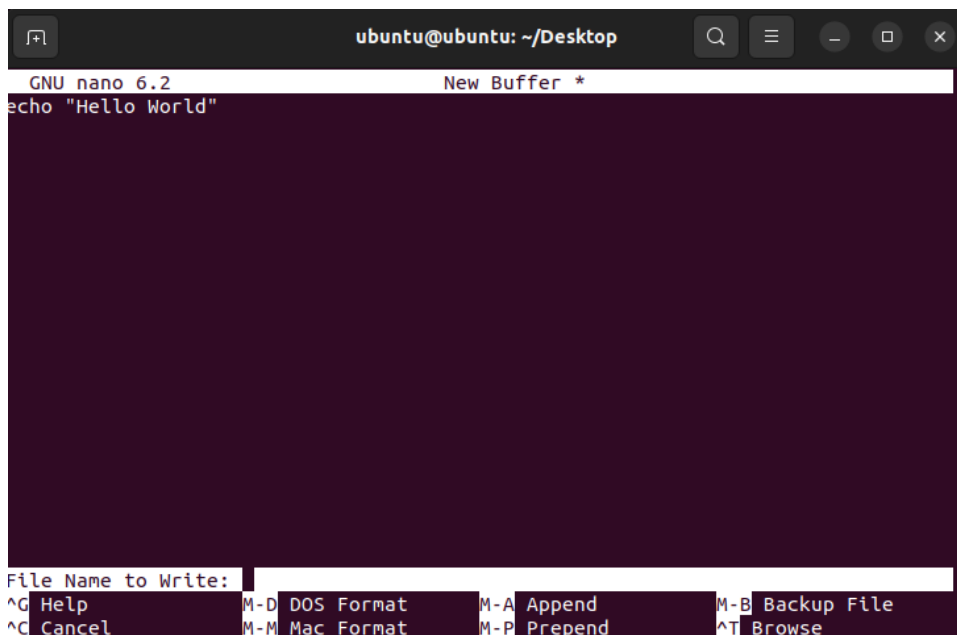
The screenshot shows a terminal window with the title 'ubuntu@ubuntu: ~/Desktop'. Inside, the nano 6.2 text editor is open, editing a file named 'New Buffer *'. The first line of the file contains the text 'echo "Hello World"'. The bottom status bar of the nano editor displays a welcome message and a grid of keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, and ^_ Go To Line.

Aby wyjść i zapisać plik naciskami CTRL+X, wyświetla nam się kolejne okienko:



This screenshot shows the same nano 6.2 editor window, but now it is prompting the user to save the changes. At the bottom of the editor, the text 'Save modified buffer?' is displayed. Below this prompt, there are three options: 'Y Yes', 'N No', and '^C Cancel'. The 'Y' option is currently selected.

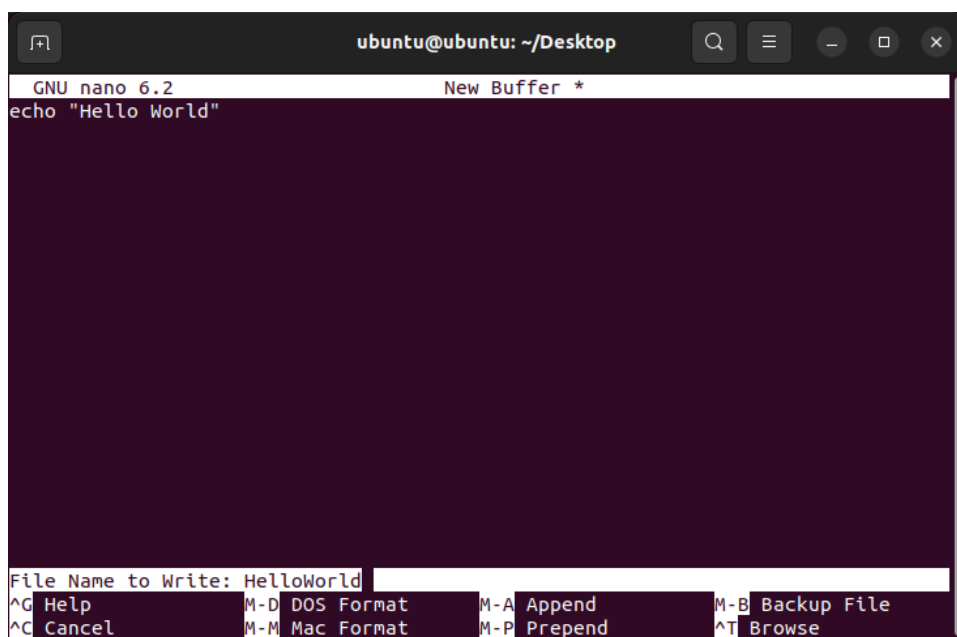
Naciskamy Y:



```
ubuntu@ubuntu: ~/Desktop
GNU nano 6.2 New Buffer *
echo "Hello World"

File Name to Write:
^G Help      M-D DOS Format  M-A Append    M-B Backup File
^C Cancel    M-M Mac Format  M-P Prepend   ^T Browse
```

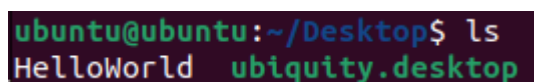
Po czym wpisujemy nazwę jaką chcemy nadać skryptowi, w moim przypadku nazwałem skrypt nazwą HelloWorld:



```
ubuntu@ubuntu: ~/Desktop
GNU nano 6.2 New Buffer *
echo "Hello World"

File Name to Write: HelloWorld
^G Help      M-D DOS Format  M-A Append    M-B Backup File
^C Cancel    M-M Mac Format  M-P Prepend   ^T Browse
```

Nowy plik utworzy się w folderze, w którym jest użytkownik



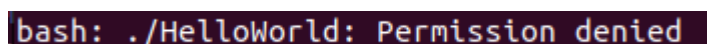
```
ubuntu@ubuntu:~/Desktop$ ls
HelloWorld  ubiquity.desktop
```

Aby uruchomić ten skrypt musimy wpisać jego nazwę oraz położenie w terminal



```
ubuntu@ubuntu:~/Desktop$ ./HelloWorld
```

Po pierwszym uruchomieniu wyświetli nam się komunikat:



```
bash: ./HelloWorld: Permission denied
```

Musimy więc przekazać sobie pozwolenie:

```
ubuntu@ubuntu:~/Desktop$ chmod +x ./HelloWorld
```

Teraz próbując uruchomić otrzymamy:

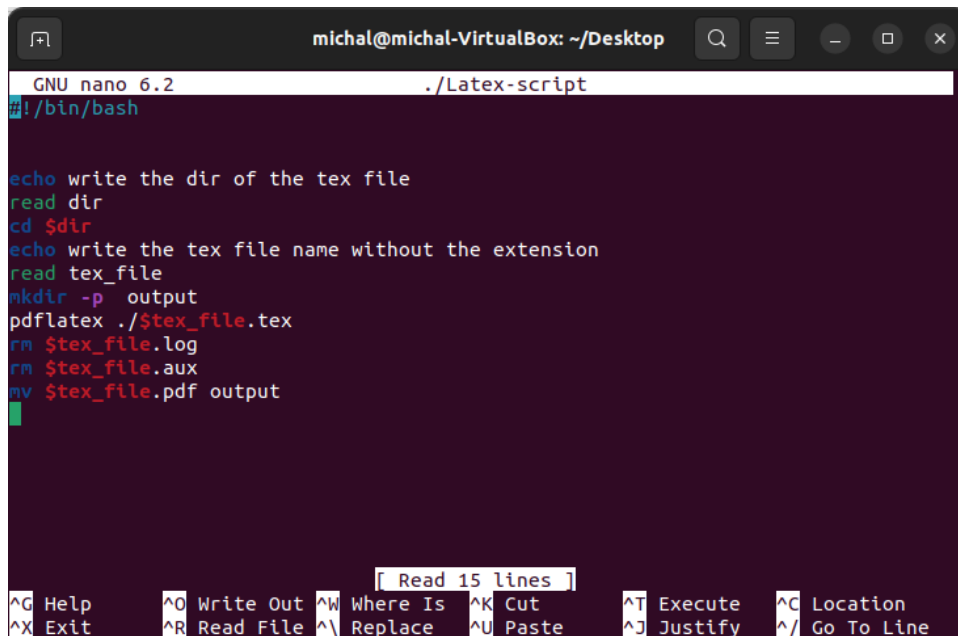
```
ubuntu@ubuntu:~/Desktop$ ./HelloWorld  
Hello World
```

Zad 4

Zad 4

Na początku trzeba pobrać plik test.tex na maszynę wirtualną

Następnie trzeba przystąpić do pisania skryptu za pomocą komendy nano. W skrypcie który napisałem dodałem również interakcje z użytkownikiem, dając możliwość wybrania folderu w którym znajduje się plik, jak i również jego nazwę. Końcowy skrypt wygląda następująco:



```
GNU nano 6.2                               ./Latex-script
#!/bin/bash

echo write the dir of the tex file
read dir
cd $dir
echo write the tex file name without the extension
read tex_file
mkdir -p output
pdflatex ./${tex_file}.tex
rm $tex_file.log
rm $tex_file.aux
mv $tex_file.pdf output
```

[Read 15 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^I Replace	^U Paste	^J Justify	^_ Go To Line

Na początku za pomocą polecenia echo pytamy się użytkownika o to, gdzie znajduje się folder zawierający plik oraz o nazwę pliku, następnie za pomocą polecenia read odczytujemy dane które podał użytkownik. Po odczytaniu danych program przechodzi do folderu wybranego przez użytkownika za pomocą polecenia cd i tworzy folder “output” za pomocą komendy mkdir po czym odczytuje plik .tex i tworzy plik .pdf za pomocą komendy pdflatex.

Na koniec program usuwa pliki .log oraz .aux za pomocą komendy rm po czym za pomocą komendy mv przenosi plik .pdf do folderu output.