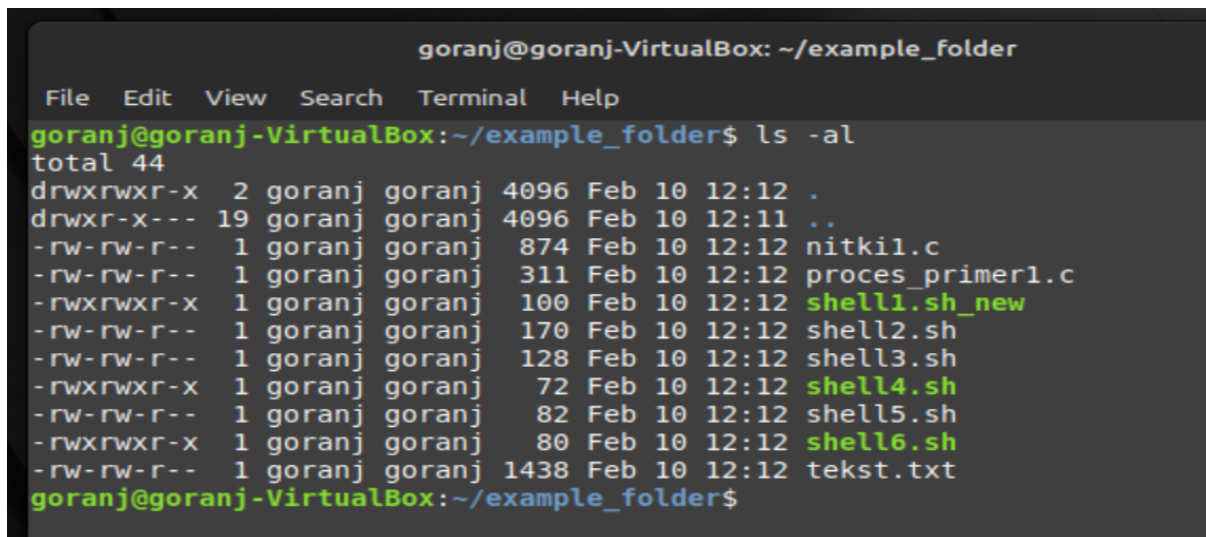


Оперативни системи

14.2.2025

Задача 1: Потребно е да се напише Python скрипта за работа со обработка на наредби. Скриптата добива на влез (аргумент од командна линија) текстуална датотека во која е сместена содржина на „ls -al“ наредбата од некој директориум. Во секоја линија има податоци за една датотека (посебна линија за посебна датотека). Доколку датотеката е со екстензија „.c“ тогаш на екран се пишува наредба за компајлирање на датотеката (како извршно име се сместува името на датотеката без екстензијата) а потоа (во нов ред) се печати наредба за извршување на програмата. Доколку датотеката е со екстензија „.sh“ и доколку истата не е со привилегии за извршување, тогаш прво се печати наредба за поставување на привилегии за извршување а потоа се извршува истата. Ова се прави за секоја датотека (линија) во текстуалната датотека. Во продолжение е пример за компајлирање, поставување на привилегии за извршување и повикување на програмите/скриптите.

Пример:



```
goranj@goranj-VirtualBox: ~/example_folder
File Edit View Search Terminal Help
goranj@goranj-VirtualBox:~/example_folder$ ls -al
total 44
drwxrwxr-x  2 goranj goranj 4096 Feb 10 12:12 .
drwxr-x--- 19 goranj goranj 4096 Feb 10 12:11 ..
-rw-rw-r--  1 goranj goranj  874 Feb 10 12:12 nitki1.c
-rw-rw-r--  1 goranj goranj  311 Feb 10 12:12 proces_primer1.c
-rwxrwxr-x  1 goranj goranj  100 Feb 10 12:12 shell1.sh_new
-rw-rw-r--  1 goranj goranj  170 Feb 10 12:12 shell2.sh
-rw-rw-r--  1 goranj goranj  128 Feb 10 12:12 shell3.sh
-rwxrwxr-x  1 goranj goranj   72 Feb 10 12:12 shell4.sh
-rw-rw-r--  1 goranj goranj   82 Feb 10 12:12 shell5.sh
-rwxrwxr-x  1 goranj goranj   80 Feb 10 12:12 shell6.sh
-rw-rw-r--  1 goranj goranj 1438 Feb 10 12:12 tekst.txt
goranj@goranj-VirtualBox:~/example_folder$
```

На екран:

```
gcc nitki1.c -o nitki1
./nitki1

gcc proces_primer1.c -o proces_primer1
./proces_primer1

chmod +x shell2.sh
./shell2.sh

chmod +x shell3.sh
./shell3.sh

./shell4.sh

chmod +x shell5.sh
./shell5.sh

./shell6.sh
```

Задача 2: Да се напише Shell скрипта која како аргумент од командна линија добива име на директориум (доколку не добие тогаш го зема тековниот директориум). Shell скриптата треба да ја изврши наредбата `ls -al <ime_na_direktorium>` и резултатот да го запише

во датотека. Потоа, да ја повика Python скриптата од задача 1 на која ќе и ја прати датотеката како аргумент од командна линија. Python скриптата како резултат враќа низа од наредби кои Shell Скриптата треба да ги изврши. Доколку наредбата е наредба за компајлирање и доколку компајлирањето не врати празен стринг, тогаш следната наредба (следната линија) за извршување на компајлираната програма не се извршува.

Задача 3: Да се напише програма во C програмскиот јазик за работа со процеси, нитки и IPC. Програмата како аргументи од командна линија добива клуч за Shared memory и име на датотека за мемориско мапирање. Процесот креира дете процес кои двата комуницираат преку Shared memory. Дете процесот мемориски ја мапира датотеката чија содржина треба да се пребара со 26 нитки. Дете процесот ги креира 26те нитки, притоа секоја нитка пребарува по една буква (во мапираната датотека) од ASCII табелата (без разлика дали е мала или голема), така што првата нитка ја пребарува буквата „a“ втората буквата „b“ и т.н. Откако ќе завршат со пребарување нитките, дете процесот, преку Shared memory му кажува на родител процесот дека нитките завршиле и резултатот од пребарувањето се запишува во shared memory. Родител процесот чека нитките на детето да завршат со извршување а потоа резултатот од пребарувањето, родител процесот го печати на екран.

Задача 4: Да се напише драјвер за комуникација со уред. Уредот има меморија од 10KB, поделена во два дела (6 KB примарна меморија се за читање и запишување, 4 KB се дополнителна меморија). Податоците во меморијата се запишуваат од лево на десно и доколку нема слободна меморија, не се запишува ништо. Доколку корисникот се обиде да запише N Бајти меморија, тие ги запишува во слободниот простор во кој нема веќе запишано меморија. Доколку има помалку од N бајти слободна меморија, тие податоци воопшто не се запишуваат во уредот. Читањето од уредот се прави од лево на десно, така што не може да се прочита повеќе од што има запишано во меморијата. Доколку се обиде корисникот да запише во драјверот 3 пати, а притоа не може да се запише затоа што нема доволно меморија, после третото неуспешно запишување заради немање доволно меморија, почнува да се користи дополнителната меморија.