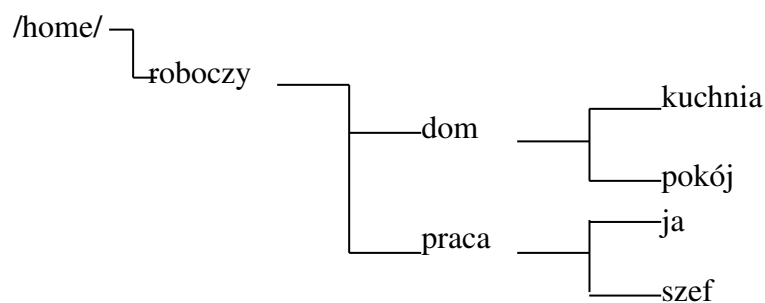


1. Utworzyć następującą strukturę katalogów:



2. Korzystając z edytora Vi utworzyć plik o nazwie ala w katalogu roboczy. Następnie przekopiować go do wszystkich podkatalogów.

Podstawowe opcje edytora vi :

a – przejście do trybu wprowadzania
 dd – usunięcie całego wiersza
 yy – kopiowanie aktualnego wiersza do bufora
 p – wstawienie tekstu z bufora
 u – cowa ostatnią zmianę
 :wq – zapisuje zmiany i zamyka program
 :q! – kończy prace bez zapisywania

3. Stworzyć plik o nazwie pomoc w którym znajdzie się pomoc na temat 5 wybranych poleceń, następnie nadać mu prawo do odczytu tylko przez właściciela.
4. Utworzyć dowiązanie do katalogu roboczy znajdujące się w katalogu domowym pod nazwą zadanie.
5. Ustawić dla katalogu praca pełne prawa dostępu dla wszystkich użytkowników, dla katalogu dom prawo odczytu i wykonania przez wszystkich użytkowników.
6. Utworzyć dowiązanie do pliku ala znajdujące się w katalogu roboczym.
7. Zapisać pięć pierwszych wierszy z pliku pomoc w pliku headpomoc, oraz dwa ostatnie kilobajty pliku pomoc w pliku tailpomoc
8. Posortować plik pomoc i wyświetlić go porcjami na ekranie
9. Zapisać posortowaną zawartość katalogu home w pliku dom.za
10. Zamontować dyskietkę i przekopiować na nią plik pomoc następnie ją odmontować
11. Usunąć katalog roboczy

mkdir – tworzy katalog
 rmdir – usuwa katalog
 chmod – ustala prawa dostępu dla plików
 (rwx – odczyt, zapis, wykonanie)
 (u – właściciel, g – grupa, o – inni)
 ls – wyświetla zawartość katalogów
 (ls -l wyświetla m.innymi prawa dostępu)
 dowolne polecenie z opcją **-help** wyświetli pomoc dla tego polecenia
 pwd – wyświetla ścieżkę bieżącego katalogu
 > - przekierowuje dane na wyjście np. do pliku
 >> - przekierowuje dane na wyjście np. do pliku i dopisuje je na końcu
 < - przekierowuje dane na wejście
 du – wyświetla ilość zajętego miejsca na dysku w odniesieniu do plików
 cp – kopiowanie plików
 mv – przenoszenie
 cat – wyświetla zawartość plików
 more - wyświetla zawartość plików porcjami
 head -n 5 - z tą opcją wyświetla pięć pierwszych wierszy z pliku, -c 5k – wyświetla pierwsze 5kB pliku)
 tail – to samo co wyżej tylko ostatnie
 cmp – porównuje pliki (-c wyświetlenie różniących ich znaków)
 rm – usuwa pliki
 ln – tworzy dowiązania do plików lub katalogów (dwie nazwy dla jednego katalogu lub pliku)
 find – przeszukuje katalogi w poszukiwaniu pliku
 (-user root szuka plików należących do root'a)
 grep (poszukuje w pliku wyr. lub wzorców
 np. grep -b wyświetla pozycję miejsca)
 | - potoki
 sort – sortuje np. (cat ala |sort sortuje plik ala)
 mount – montuje system plików np.
 mount /dev/fd0 /mnt/floppy –montuje system plików dyskietki do katalogu floppy
 umount – odmontowuje np. umount /mnt/floppy
 ~/katalog macierzysty

12. Zarchiwizuj plik **ala** programem tar (ala.tar), następnie archiwum ala.tar skompresuj używając formatu gzip (ala.tar.gz). następnie przekopiuj plik ala.tar.gz i rozpakuj go.
tar -c nazwaarchiwa.tar nazwaźródła (-x rozpakowuje, -c pakuje)
gzip -c nazwaźródła > nazwaarchiwa.gz (-d dekompresuje (lub gunzip), -c pakuje)

13. Sprawdzić rodzaj używanej powłoki, której używasz (echo \$SHELL), zmień tymczasowo powłokę na **cs** (/bin/csh), następnie wyjdź **exit**. (zmień też powłokę na **tcsh**)

14. Napisz skrypt, który wyświetli powitanie np. *Witaj, mamy piękny dzień!!!!!!*

```
#!/bin/basch
```

```
# komentarz (#!/bin/basch - wskazuje na rodzaj shella (powłoki) w jakim skrypt ma być wykonany)
```

```
echo „. Witaj, mamy piękny dzień!!!!!!”
```

Aby uruchomić skrypt należy nadać mu prawo wykonywalności:

Chmod +x nazwaskryptu

Uruchamiamy **./nazwaskryptu** jeśli katalog nie jest dopisany do zmiennej **Path**.

15. Napisz skrypt, w którym zostanie utworzona zmienna x, następnie wartość tej zmiennej zostanie wydrukowana, pojawi się też data oraz aktualny katalog, ponadto symbol zmiennej z jej nazwą (\$x).

```
#!/bin/basch
```

```
x=2
```

```
echo „Wartość zmiennej x to $x”
```

```
echo „Dzisiejsza data to `date`”
```

```
echo „Znajdujemy się w katalogu `pwd`”
```

```
echo „Taki zapis nie wyświetli wartości zmiennej x: \ $x”
```

\$ wskazuje na nazwę zmiennej
\
znak maskujący
`` odwrotny apostrof zacytowanie polecenia

16. Utwórz skrypt, który wyświetli liczby, następnie doda i odejmie. Liczby podane są jako parametry.

\$1, \$2 – parametry

\$@ - pokazuje wszystkie parametry z jakimi został uruchomiony skrypt