

WSOpracowane ;-)

1) Co oznaczają(w odniesieniu do komputerowego systemu operacyjnego) słowa :

a) Wielodostępny(multiuser).

W systemie jednocześnie może pracować wielu użytkowników.

b) Wielozadaniowy(multitask).

W systemie można pracować z wieloma aplikacjami jednocześnie.

W tym samym czasie wielu użytkowników może wykonywać kilka zadań na tym samym komputerze.

c) Pracujący z podziałem czasu(time sharing).

Komputer może wykonywać kilka zadań jednocześnie.

d) Które z tych słów są prawdziwe w odniesieniu do systemu Unix?

Wszystkie :)

2) Czy administrator może odczytać hasła użytkowników? Uzasadnij.

Nie. Hasło jest 'szyfrowane' jednokierunkowo i dopiero w takiej postaci składowane w maszynie.

3) Jakim poleceniem można sprawdzić rodzaj i wersję systemu operacyjnego w którym aktualnie pracujesz?

```
$ uname -a
```

```
Linux blade-runner.wsisiz.edu.pl 2.6.14 #1 SMP Wed Nov 2 12:47:18 CET 2005 i686  
i686 i386 GNU/Linux
```

4) Jesteś w swoim katalogu osobistym :

a) Podaj najkrótsze polecenie, którym możesz utworzyć podkatalog pod1/pod2/pod3 Twojego podkatalogu osobistego.

```
$ mkdir -p pod1/pod2/pod3
```

b) Jakie są inne metody?

```
$ mkdir pod1
```

```
$ cd pod1
```

```
$ mkdir pod2
```

```
$ cd pod2
```

```
$ mkdir pod3
```

lub

```
$ mkdir pod1 pod1/pod2 pod1/pod2/pod3
```

lub

```
$ mkdir pod1
```

```
$ mkdir pod1/pod2
```

```
$ mkdir pod1/pod2/pod3
```

c) Podaj polecenia tworzące w nich po jednym pliku o nazwach odpowiednio plik1, plik2, plik3(mogą być puste).

```
$ touch pod1/plik1 pod1/pod2/plik2 pod1/pod2/pod3/plik3
```

5) Podaj polecenia, którymi można sprawdzić zawartość katalogu np. o nazwie "mona".

```
$ ls mona
```

lub \$ dir mona

Wymień inne możliwości :

a) Bez uwzględnienia plików ukrytych.

```
$ ls mona
```

*ls domyślnie nie pokazuje plików ukrytych.

b) Z uwzględnieniem plików ukrytych.

```
$ ls -a mona
```

c) Jak można odczytać prawa dostępu do katalogu pod2 (z pytania 4)

```
$ ls -l pod1/ | grep pod2
```

```
lub $ ls -ld pod1/pod2
```

6) Podaj polecenie ustawiające prawa pliku plik2 na :

a) Wszystkie dla właściciela, czytanie i pisanie dla grupy do której należy ten plik oraz żadne dla pozostałych użytkowników.

```
$ chmod u=rwx,g=rw,o-rwx plik2
```

```
lub $ chmod 760 plik2
```

b) Opisz krótko znaczenie symboli *r*, *w*, *x* w stosunku do katalogu.

r - umożliwia przeglądanie zawartości katalogu (*ls*)

w - umożliwia modyfikowanie zawartości katalogu (*rm*, *cp*, *mv*, *mkdir*, *rmdir*, *ln*)

x - uruchomiony program lub wydane polecenie ma dostęp do pliku w katalogu (*cd*, *ll* nazwapliku)

c) Czy ktoś poza właścicielem ma prawo zmieniać prawa dostępu. Jeśli tak to kto?

Tylko właściciel lub administrator mogą zmieniać prawa dostępu.

7) Podaj jak określić odpowiednią zmienną shella, aby znak zachęty zawierał identyfikator użytkownika i nazwę katalogu bieżącego - wykorzystując odczyt wartości pewnych shella.

```
$ PS1="$UID($USER) @$PWD"
```

Jakie są inne sposoby?

```
$ PS1="\u@\w"
```

```
$ PS1="\u@W"
```

8) Podaj poszczególne kroki prowadzące do utworzenia skryptu o nazwie *skr1*, którego wykonanie wyśle na ekran :

"To jest skrypt Darka " i utworzy w Twoim katalogu osobistym plik o nazwie Twoje imię.

```
$ cat > skr1
```

```
echo To jest skrypt Darka
```

```
touch Maciej
```

```
CTRL + D
```

```
$ chmod u+rx skr1
```

```
$ ./skr1 lub $ . skr1
```

9) Jak najprościej dopisać do skryptu *skr1* mały fragment tekstu?

```
$ echo mały fragment tekstu >> skr1
```

a) Jakimi sposobami można wykonać skrypt i jakie prawa musi mieć wtedy plik *skr1*?

```
$ ./skr1 lub $ bash skr1 lub $ . skr1
```

Prawo do odczytu i wykonywania(*r*, *x*).

b) Jak wywołać *skr1*, tak aby został wywołany w shellu bieżącym(a następnie w potomnym)?

shell bieżący \$./skr1 lub \$. skr1

shell potomny \$ bash skr1 lub \$ \$SHELL skr1

10) Zbuduj w postaci potoku polecenie :

a) Zapisujące na dysk o nazwie "" sesję posortowaną(wg. czasu zalogowania) oraz listę posortowanych sesji. Podaj inny przykład potoku.

b) Jak wyglądałoby polecenie wyświetlające na ekranie zamiast listy jedynie liczbę otwartych sesji?

```
$ who | wc -l
```

c) Do czego służy polecenie *grep*. Użyj polecenia *grep* do wyświetlania informacji o sesjach otwartych przez użytkownika *userc22*.

grep - Global Regular Expression Printer

Polecenie *grep* wyszukuje wiersze zawierające podany wzorzec. Wzorzec jest zdefiniowany za pomocą wyrażenia regularnego(*regular expression*).

d) Jak uzyskać na ekranie listę terminali (i tylko terminali) na których utworzono sesję?

e) Sprawdź czy masz przypisany skrót klawiszowy CTRL + Z umożliwiający zawieszenie wykonania zadań w pierwszym planie. Podaj polecenie, którym to sprawdzisz.

```
$ stty -a
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>; eol2 =
<undef>; start = ^Q; stop = ^S;
susp = ^Z; rprnt = ^R; werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
```

f) Jak uzyskać informację o numerach PID oraz jobs, dla wykonywanych zadań. Podaj ciąg poleceń odpowiedzi na ekranie, związany z realizacją następującej sekwencji poleceń : uruchomienie w planie pierwszego procesu, związanego z wykonaniem polecenia sleep 1000, zawieszenie wykonania tego procesu, wznowienie go w drugim planie, bezwarunkowe usunięcie tego procesu

g) Jak uruchomić proces w tle odporny na wylogowanie się. Jak jeszcze można uruchomić proces. Podaj przykład.

11) Jakim poleceniem możesz zdobyć informacje o Twoim identyfikatorze i grupie do której należysz. Czy identyfikator może ulec zmianie w czasie sesji? Jeżeli tak to jak możesz dowiedzieć się jaki był na początku sesji a jaki jest aktualnie?

Polecenie id wyprowadza na standardowe wyjście numeryczne identyfikatory użytkownika i grupy, do której został zaliczony przy otwieraniu jego konta. Jeżeli efektywny i rzeczywisty identyfikator użytkownika nie jest identyczny, wyprowadzane są oba.

Można chwilowo uzyskać uprawnienia innego użytkownika za pomocą zmiany :

Identyfikator użytkownika - polecenie su

Identyfikator grupy - polecenie newgrp

Składnia :

```
su [-] [nazwa_użytkownika]
```

```
newgrp [nazwa_grupy]
```

su bez argumentu - wybiera użytkownika root .

newgrp bez argumentu - przywraca grupę początkową użytkownika.

exit - przywraca poprzedni identyfikator użytkownika.

```
$ id
uid=9099(wysockm0) gid=100(users) groups=100(users)
$ who am i
wysockm0 pts/1          Nov 27 21:24 (cjc127.neoplus.adsl.tpnet.pl)
$ su sikorski
$ logname
wysockm0
$ who am i
wysockm0 pts/1          Nov 27 21:24 (cjc127.neoplus.adsl.tpnet.pl)
$ id
uid=7008(sikorski)  gid=203(pion3)

$ exit

$ id
uid=9099(wysockm0) gid=100(users) groups=100(users)
```

12) Jak zorientować się czy Twój katalog osobisty przeglądany jest przez shell w trakcie poszukiwania polecenia?

```
$ echo $PATH
/usr/local/bin:/bin:/usr/bin:/sbin:/usr/sbin/:/usr/X11R6/bin:/usr/local/bin:/usr/
local/sbin:/home/informatyka/wysockm0/bin
```

a) Jak i którą zmienną środowiskową zmodyfikować, aby był on przeglądany?

```
$ PATH=$HOME
$ echo $PATH
/home/informatyka/wysockm0
```

b) Jak sobie poradzisz z wykonaniem skryptu umieszczonego w Twoim katalogu osobistym(bez wprowadzania katalogu osobistego do zmiennej PATH)?

```
$ /home/informatyka/wysockm0/skrypt
```

13) Wymień co najmniej 3 najważniejsze (wg Ciebie) opcje polecenia ls, jakich informacji dostarczają?

- a - wyświetla wszystkie pliki w katalogu, wraz z plikami ukrytymi.
- i - pokazuje numery indeksowe(i-node, i-węzły) każdego pliku. Numery te jednoznacznie identyfikują pliki w ramach danego systemu plików.
- l - listuje w długim formacie pełną informację o plikach i katalogach.
- R - listuje rekurencyjnie zawartości wszystkich katalogów(czyli wyświetla również podkatalogi).
- F - zaznacza za pomocą symbolu
 - / każdą nazwę katalogu,
 - * plik z programem binarnym
 - @ dowiązania symboliczne.
- d - wyświetla informację o katalogu, bez jego zawartości.

14) Do czego służą opcje -i -b w poleceniach mv i cp?

-i, --interactive

Pyta użytkownika czy nadpisywać pliki docelowe, które już istnieją. Jeśli odpowiedź nie zaczyna się od 'y' lub 'Y', plik jest pomijany.

-b [metoda], --backup[=metoda]

Tworzy kopie zapasowe plików, które mają zostać nadpisane lub usunięte.

*W poleceniu cp, opcja '-b' nie posiada wartości.

15) Czemu służy dowiązanie twarde(hardlink) i miękkie(symboliczne) tzw. softlink?

Dowiązania **twarde** charakteryzują się tym, że posiadają ten sam i-węzeł co wskazywany plik. W praktyce oznacza to, że dowiązanie takie zawiera pełną informację o pliku i w razie skasowania oryginalnego pliku dowiązanie nadal będzie zawierać informacje o tym pliku (jego zawartość).

Dowiązania **symboliczne** służą jedynie do wskazywania na jakiś plik. Umożliwiają one prace na danym pliku poprzez jego dowiązanie. W przypadku skasowania pliku, do którego prowadzi link symboliczny niemożliwe staje się korzystanie z tego dowiązania (nie zawiera on wówczas żadnych informacji).

a) Kiedy można zastosować dowiązanie twarde a kiedy miękkie?

Dowiązania **twarde** można zastosować do plików znajdujących się tylko na tym samym dysku i w tej samej partycji. Dowiązania **miękkie** można tworzyć do plików znajdujących się w dowolnym miejscu systemu plików.

b) Podaj polecenie tworzące dowiązanie twarde o nazwie plik1dt do pliku o nazwie plik1 i umieszczenie w katalogu pod3(układu katalogów i plików z zadania 4).

```
$ ln pod1/plik1 pod1/pod2/pod3/plik1dt
```

c) Jak można pokazać, że plik1 oraz plik1dt to fizycznie ten sam plik zapisany na dysku?

```
$ ls -li pod1
```

```
257626 -rw-r--r-- 2 wysockm0 users 9 Nov 11 15:09 plik1
```

*2 to liczba dowiązań... 257626 to numer indeksowy(i-węzeł)

```
$ ls -li pod1/pod2/pod3
```

```
257626 -rw-r--r-- 2 wysockm0 users 9 Nov 11 15:09 plik1dt
```

Numer indeksowy jest ten sam, więc są to te same pliki

*Można przejrzeć zawartość pliku. Polecenie `$ cat pod1/plik1` jest równoważne `$ cat pod1/pod2/pod3/plik1dt`

d) Podaj polecenie tworzące analogiczny link symboliczny o nazwie plik1ds.

```
$ ln -s ~/pod1/plik1 pod1/pod2/pod3/plik1ds
```

```
lub $ ln -s $HOME/pod1/plik1 pod1/pod2/pod3/plik1ds
```

"~" oznacza katalog domowy (np. /home/informatyka/wysockm0).

*W 'softlinkach' do plików trzeba podawać dokładny adres do pliku.

**Można sprawdzić gdzie jest nasz katalog domowy wpisując `$ echo $HOME` lub `$ echo ~`

```
$ ln -s pod1/pod2/pod3 bliski
```

Niepusty \$ rm -fr katalog

*Skrót od Print Working Directory

PWD

\$ 1s -1

-r--r--r--	1	ania	plon1	5632	Apr 3	14:59	plik1
			Nazwa grupy				
		Nazwa właściciela					
	Liczba dowiezań						
Prawa dostępu							
Typ pliku							

```
$ ls -ld konkretny/katalog
```

```
$ ls -alp | grep ' \./'
```

```
lub $ alias kata="ls -AF | grep '/'"
```

umask - służy do określania jakie prawa mają mieć pliki i katalogi w momencie ich zakładania

```
$ umask 140
777-140=637 takie prawa dostępu będą mieć nowo utworzone katalogi
666-140=526 takie prawa dostępu będą mieć nowo utworzone pliki
*Domyślnymi prawami dostępu dla pliku są 666 a katalogu 777.
```

20) Zapisz do pliku admin tekst przesłany Tobie przez administratora za pośrednictwem poczty.

21) Wybierz z informacji wprowadzonych przez polecenie data nazwę dnia i wydrukuj ja na ekranie dużymi literami.

```
$ date +%A | tr [a-z] [A-Z]
```

22) Przypisz zmiennej PAN swoje imię i nazwisko. W jaki sposób można zobaczyć wartość tej zmiennej? Co trzeba zrobić, aby shell potomny widział zawartość zmiennej PAN?

```
$ PAN="Maciej Wysocki"
```

```
$ echo $PAN
```

```
Maciej Wysocki
```

```
$ export PAN
```

```
$ bash
```

```
$ echo $PAN
```

```
Maciej Wysocki
```

23) W jaki sposób wyświetlisz podany komunikat : Testy #1, \$2, &3 są zrobione

```
$ echo "Testy #1, \$2, &3 są zrobione"
```

```
albo $ echo Testy \#1, \$2, \&3 są zrobione
```

24) Napisz zestaw poleceń w postaci potoku, który z pliku /etc/passwd wybierze nazwy i ścieżki do katalogu osobistego użytkowników, których identyfikatory rozpoczynają się od user.

```
$ cut -d: -f1,6 /etc/passwd | grep "^user"
```

25) Przygotuj plik raport, w którym kolejne wiersze zawierają

Polecenie, które zapisze w pliku raport.log bieżącą datę i nazwę bieżącego katalogu.

Polecenie, które dopisze do tego samego pliku listę aktualnie pracujących w systemie użytkowników.

Polecenie, które na końcu pliku raport.log umieści identyfikator użytkownika, który uruchomił plik raport.

*Na początek możemy stworzyć plik i nadać właściwe prawa \$ touch raport && chmod u+x raport

```
$ cat > raport
```

```
echo `date` `pwd` > raport.log
```

```
who >> raport.log
```

```
echo "$UID($USER)" >> raport.log
```

```
CTRL + D
```

26) Utwórz synonim polecenia o nazwie mv wymagający potwierdzenia zapisania na istniejący już plik oraz synonim dira wypisujący zawartość katalogu wraz z plikami ukrytymi.

```
$ alias mv="mv -i"
```

```
$ alias dira="dir -a" lub $ alias dira="dir -A"
```

27) W których plikach w Twoim katalogu osobistym występuje słowo 'wiersz' pisane małymi lub dużymi literami?

```
$ grep -ils wiersz *
```

28) Zmodyfikuj plik .profile tak, aby na początku każdej sesji była usuwana zawartość pliku, w którym zapisywana jest historia poleceń.

```
$ echo rm -f $HISTFILE >> .profile
```

29) Utwórz katalog o nazwie schowek o następujących prawach dostępu :

Ty masz wszystkie prawa dostępu do katalogu.
Członki Twojej grupy mogą przeglądać, ale nie mogą modyfikować jego zawartości.
Inni użytkownicy nie mają żadnych praw do tego katalogu.

```
$ mkdir schowek -m 750
lub
$ mkdir schowek
$ chmod u=rwx,g=rx,o-rwx schowek
```

30) Jak należy ustawić maskę praw dostępu, aby atrybuty nowotworzonych plików i katalogów były :

Dla pliku : rw-r--r--
Dla katalogu : rwxr-xr--

Dla pliku \$ umask 022
*666-022=644
Dla katalogu \$ umask 023
*777-023=754

31) Nie używając edytora tekstu utwórz plik info zawierający wyrazy : 'dzisiaj piszę test!'

```
$ echo dzisiaj piszę test\! > info
Używając edytora vi utwórz plik lot zawierający tekst :
'W lutym pojadę na narty do Szczyrku
Muszę jeszcze kupić narty.'
$ vi lot
```

32) W pliku lot

a) dopisz na końcu pierwszego wiersza wyrazy "lub Zakopanego"
b) zastąp litery "y" literami "xyz"
\$ cat lot | tr "y" "xyz" > lot
c) wyraz "narty" przekopiuj na końcu pliku 50 razy
d) ponumeruj wiersze

33) W pliku lot zamień znaki spacji na znaki nowej linii i wynik zapisz w pliku lot.m

```
$ cat lot | tr " " "\n" > lot.m
```

34) Zbuduj potok zliczający liczbę użytkowników nie korzystających z powłoki bash oraz zapisujący identyfikatory tych użytkowników w kolejności alfabetycznej do pliku.

```
$ cut -d: -f7 /etc/passwd | grep -v "/bin/bash$" | wc -l | cut -d: -f1,7
/etc/passwd | grep -v "/bin/bash$" | cut -d: -f1 | sort > plik
albo
$ cat /etc/passwd | grep -v "/bin/bash$" | wc -l | cut -d: -f1,7 /etc/passwd |
grep -v "/bin/bash$" | cut -d: -f1 | sort > plik
```

35) W jaki sposób wyświetlisz komunikat : Zadania \$1, '2' i \$11 są już za nami?

```
$ echo "Zadania \$1, '2' i \$11 są już za nami"
albo $ echo Zadania \$1, \'2\' i \$11 są już za nami
```

36) Wyświetl wszystkie wiersze w pliku /etc/passwd które zawierają napis user. Zachowaj ten wynik w pliku o nazwie uzyt.

```
$ cat /etc/passwd | grep user > uzyt
```

a) Policz wiersze pliku uzyt.

```
$ wc -l uzyt
```

37) Wybierz dzień i rok z informacji wprowadzanych przez polecenie date.

```
Dzień $ date +%d
Rok $ date +%Y
```

38) Wyszukaj w całym systemie plików te z nich które mają więcej niż jedno dowiązanie.

```
$ ls -lR | grep -v ' 1 '
```

39) Przypisz zmiennej DANE nazwisko użytkownika, który otworzył sesję pracy na terminalu, z którego wydano to polecenie.

```
$ DANE=`finger $USER | grep -i login | cut -d: -f3 | cut -d ' ' -f3`
$ echo $DANE
Wysocki
```

40) Wyślij komunikat "Czas na przerwę" do użytkownika mój.

```
$ write mój
Czas na przerwę
CTRL + D
```

41) Utwórz synonim o nazwie oni, którego zadaniem będzie wyświetlanie alfabetycznego spisu użytkowników mających aktualnie otwarte sesje w systemie.

```
$ alias oni="who | sort | cut -c1-8"
albo $ alias oni="who | sort | cut -d ' ' -f1"
```

42) Po wydaniu polecenia exit na ekranie pojawił się komunikat "You have stopped jobs". Co oznacza ten komunikat i co powinienś zrobić w tym przypadku?

Oznacza to, że nie zostały jeszcze zakończone procesy uruchomione w tle.
Można wtedy :
- zakończyć zadania.
- ponownie wydać polecenie exit - procesy automatycznie zostaną zakończone.

*W shellu ksh można zadanie uruchamiane w drugim planie poprzedzić poleceniem nohup. Wtedy nie zakończone procesy będą wykonywane również po zakończeniu sesji. Są one wtedy adoptowane przez proces init.

43) Zmodyfikuj polecenie date tak aby jego wyjście miało postać :

```
Halo ! Bieżąca data : mm\dd\yy
Bieżący czas : gg:MM:ss
$ date +"Halo \"%t\"Bieżąca data : \"%m\\%d\\%y%n\"Bieżący czas : \"%H\":\"%M\":\"%S"
```

44) Jak utworzyć dwa pliki ala.st i ala.nd powiązane z zawartością pliku ala.?

```
$ ln -s ala. ala.st
$ ln -s ala. ala.nd
```

45) Wydano polecenie umask 123. Jakie będą atrybuty nowotworzonych plików i katalogów? Jakimi poleceniami można to sprawdzić? Czy polecenie umask 4123 jest poprawne?

```
$ umask 123
Pliki :
666-123=5|4|3
tzn. nowe pliki będą posiadać takie atrybuty : r-x|r--|-wx
Katalogi :
777-123=6|5|4
rw-|r-x|r--
Aby sprawdzić atrybuty można użyć polecenia $ ls -l
```

\$ umask 4123 nie jest poprawne. Bash nam powie "octal number out of range". 'Czwarta cyfra' w umask nie może być różna od zera...

46) Zmień znak gotowości shella aby wyświetlić :

a) Nazwę użytkownika i numer wykonywanego polecenia.

```
$ export PS1="\u \#"
```

```
wysockm0 7
```

b) Napis 'Co dalej IDENTYFIKATOR \$'.

```
$ export PS1="Co dalej $UID($USER) \ $"
```

```
Co dalej 9099(wysockm0) $
```

47) Używając jednego polecenia utwórz w katalogu osobistym następujące katalogi : kk/Z/B1, kk/Z, kk/Z/B2, kk/Z/B1/C1

```
$ mkdir -p kk/Z/B1/C1 kk/Z/B2
```

48) Jakim poleceniem wyświetlisz właściwości powłoki?

```
$ set -o
```

```
allexport          off
```

```
braceexpand       on
```

```
emacs             on
```

```
errexit           off
```

a) Jak włączysz 'właściwość'?

```
$ set -o allexport
```

```
$ set -o
```

```
allexport          on
```

b) Jak ją wyłączysz?

```
$ set +o allexport
```

```
$ set -o
```

```
allexport          off
```