

**Resene priklady a poznamky k predmetu BI-UOS 2013/14.**  
**Nezarucuji spravnost ani uplnost textu.**

## **PRIPRAVA K 1. TESTU**

[https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos\\_vzor\\_1\\_test\\_1\\_zs1213](https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos_vzor_1_test_1_zs1213)

**Spustili jsme prikaz `jmeno="echo karel"` Co pote vypise kazdy z techto prikazu:**

- 1) `echo "jmeno"`**
- 2) `echo "$jmeno"`**
- 3) `echo ' $jmeno'`**
- 4) `echo '\ $jmeno'`**
- 5) `echo `jmeno``**
- 6) `echo ` $jmeno``**

- 1) jmeno
- 2) echo karel
- 3) \$jmeno
- 4) \ \$jmeno
- 5) bash: jmeno: command not found
- 6) karel

**Definujte promennou U, do ktere date nejake uzivatelske jmeno (napr. bill) a promennou S, do ktere date adresu nejakeho serveru (napr. linux.microsoft.com)**

U=bill

S=linux.microsoft.com

**Vytvorte prikaz, který provede pripojeni pres ssh, jako uzivatelske jmeno pouzije obsah promenne U a jako adresu serveru obsah promenne S**

ssh "\$U"@"\$S"

**Nastavte, aby tento prikaz mohli vykonavat i potomci aktualniho procesu (= videli obsah promennych U a S)**

export {U,S}

**Mame prikaz date jehož vystup vypada takto: Wed Oct 5 17:29:06 CEST 2011 (Napoveda: prikaz `wc -w` vypise pocet slov retezce, který prijde na jeho standardni vstup).**

**1) Napiste prikaz, který vypise pocet slov z prikazu date**

date | wc -w

**2) Napiste prikaz, který vypise dvojnásobek poctu slov z prikazu date**

echo \$((2\*`date | wc -w`))

**Napiste prikaz, který vypise datum ve formátu: POZOR - 11. dne 10. mesice roku 2011 nastane konec sveta. Nic nedelam.**

1) `echo "POZOR - `date +%d`. dne `date +%m`. mesice roku `date +%Y` nastane konec sveta. Nic nedelam."`

2) `date '+POZOR - %d. dne %m. mesice roku %Y nastane konec sveta. Nic nedelam.'`

3) `date +"POZOR - %e. dne %m. mesice roku %Y nastane konec sveta. Nic nedelam."`

**Nastavte, aby se tento prikaz spustil po zadani prikazu ls (staci pro soucasnou session)**

1) `alias ls='echo "POZOR - `date +%d`. dne `date +%m`. mesice roku `date +%Y` nastane konec sveta. Nic nedelam."'`

2) `alias ls="date +"POZOR - %e. dne %m. mesice roku %Y nastane konec sveta. Nic nedelam.\""`

[https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos\\_test\\_2013-10-14-1](https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos_test_2013-10-14-1)

**Do promenne KONEC ulozte pocet minut zbyvajicich do nasledujici hodiny. Pote text vypiste ve formátu <X> kde X je hodnota promenne KONEC.**

1) `KONEC=$((60-10#$((date +%M)))`

`echo "<${KONEC}>"`

2) `KONEC=$((60-`date +%M`))`

`echo "<${KONEC}>"`

**Zobrazte manualovou stranku sekce 2 prikazu unlink a pouzijte strankovac less. Strankovac se nesmi zmenit pro dalsi prikazy.**

man -s 2 unlink | less  
(rozdíl less / more: less dovoluje i pohyb zpatky)

**Do souboru shell.url ulozte URL <http://cs.wikipedia.cz/.../>, pote pomoci wget stahnete stranku na adrese ulozene v souboru shell.url a ulozte ji do souboru shell.html. Poznamka: prikazu wget predate adresu jako prvni parametr. Prikaz wget vypisuje stazeny obsah na standardni vystup.**

```
1) echo http://cs.wikipedia.org/wiki/Hlavn%C3%AD_strana > shell.url
   wget `cat shell.url` -O shell.html
2) echo http://cs.wikipedia.org/wiki/Hlavn%C3%AD_strana > shell.url
   wget -i shell.url -O shell.html
```

**Najdete vsechny soubory z podslozky data/, ktere v nazvu obsahuji hodnotu promenne NECO. Tyto soubory nasledne ulozte na server fray1.fit.cvut.cz do adresare /tmp. Poznamka: Hodnotu promenne NECO muzou soubory obsahovat jak na zacatku, tak na konci nebo uprostred nazvu.**

```
scp data/*$NECO* "$USER"@fray1.fit.cvut.cz:/tmp
```

**Vytvorte alias d, ktery vypise aktualni datum ve formatu YMD-hm**

```
alias d='date +%Y%m%d-%H%M'
```

**Vypiste pocet radku o jedno mensi nez ma soubor /etc/passwd**

```
echo $((`cat /etc/passwd | wc -l`-1))
```

**Do promenne S ulozte text `"*1. test"` (bez uvozovek vctne mezery) pote do promenne T ulozte promennou S a vypiste T na standardni vystup.**

```
S='*1. test*'; T="$S"; echo "$T"
```

[https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos\\_test\\_2013-10-14-2](https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos_test_2013-10-14-2)

**Máte proměnnou prostředí A v níž je uložen řetězec a proměnnou prostředí B v níž je uloženo jméno souboru. Zapište hodnotu uloženou v proměnné A do souboru, který má uložené jméno v proměnné B. Poté proměnnou A zrušte. Pokud v souboru již něco bylo obsah přepište.**

```
echo "$A" > $B; unset A
```

**Vypoctete dekadicky soucet cisel 0xDEAD a 0x13**

```
echo "$((0xDEAD + 0x13))"
```

**Vypiste pocet prvku s priponou .c z nadrazeneho adresare domovskeho adresare**

```
ls -l ~/./*.* | wc -l
```

**Do proměnné WIKI uložte adresu <http://en.wikipedia.org/wiki/>. Poté pomocí příkazu wget stáhněte do souboru shell.html stránku <http://wiki/xxxxxxx> kde wiki reprezentuje obsah proměnné WIKI.**

```
WIKI=http://en.wikipedia.org/wiki/
wget ${WIKI}xxxxxxx -O shell.url
```

[https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos\\_test\\_2013-10-15-1](https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos_test_2013-10-15-1)

**Zajistete, aby bylo mozne spoustet soubory z adresare bin umisteneho v domovskem adresari pouze zadanim jejich jmena. Ulozte nastaveni pro dalsi prihlaseni, pricemz ostatni nastaveni musi zustat funkcní.**

```
echo "PATH=$PATH:~/bin" >> ~/.bashrc;
export PATH
```

**Do promenne RSEC ulozte pocet vterin zbyvajici do konce minuty. Vypiste na standardni vystup stylem \$(S), kde S je pocet vterin**

```
RSEC=$((60 - 10#`date +%S`))
echo "\${RSEC}"
```

**Spustte soubor excel z adresare bin, ktery je umisten v nadrazenem adresari. Programu predejte argument peti mezer.**

```
./bin/excel ' ' ' ' ' '
```

**Do promenne Z ulozte seznam souboru oddeleny mezerami z adresare /usr/bin zacinajicich na pismeno a.**

```
1) Z=`ls /usr/bin/a*`          ..vypisuje zvlast na radky
```

( `` se rovna \$( ) )  
2) Z=`echo ls /usr/bin/a\*` ..dohromady

**Uložte všechny soubory s příponou .cpp a .c z podadresáře data na server fray1.fit.cvut.cz do stejného adresáře, jako je váš aktuální.**

Scp ./data/\*.c ./data/\*.cpp "fray1.fit.cvut.cz:\$PWD"  
(pwd je příkaz, \$PWD proměnná)

**Nastavte manuálový stránkováč na program wc a pak pomocí něho v této session zobrazte manuál sekcce 3c funkce printf.**

Export PAGER=wc;  
man -s 3C printf

**V proměnné CMD je uloženo "/bin/pwd" v proměnné subdir je uloženo „data“. Uložte výstup „PATH++===zde bude hodnota proměnné CMD/zde bude hodnota subdir“ do souboru drig.cfg ve vašem aktuálním adresáři. (++ odpovídá mezerám, vyhrad'te pro ně kolonky) Například pokud bychom měli proměnné HOME s hodnotou /usr/dict/ a subdir s hodnotou /data, v koncovém souboru by to mělo vypadat takto: PATH ++ ==+/usr/dict/data.  
echo "PATH++===\${CMD}/\${subdir}" > drig.cfg ..pry mala chyba,nvim kde?**

**V souboru words, který se nachází ve vašem aktuálním adresáři je seznam slov, která jsou vypsaná pod sebou na jednotlivé řádky. Vložte do proměnné S všechna tato slova na jednu řádku oddělená alespoň jednou mezerou.**

S=`cat words | tr '\n' ' '`

[https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos\\_test\\_2013-10-15-2](https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos_test_2013-10-15-2)

**Do proměnné PATH uložte cestu k aktuálnímu adresáři pomocí pwd. Potom změňte složku do /bin a použijte příkaz ls (je v aktuálním adresáři). Poté se vraťte do předchozího adresáře pomocí PATH.**

PATH=`pwd`; cd /bin; ./ls; cd "\$PATH"

**Nastavte alias t, který vypíše datum ve formátu „\* HH:MM:SS“ včetně mezery a hvězdy. Zajistěte, aby byl alias použitelný i při dalším přihlášení.**

alias t='date +"\* %H:%M:%S"'; alias t >> ~/.bashrc

**V proměnné USER je uživatelské jméno, v SERVER název serveru. Zkopírujte ze serveru z adresáře, který je stejný jako aktuální adresář všechny soubory končící na .pdf, do aktuálního adresáře na místním počítači.**

Scp "\${USER}@\${SERVER}:\${PWD}/\*.pdf" ./

**Vytvořte alias „d“, pro výpis aktuálního data. Dále tento alias použijte ve výpisu následujícího řetězce: “[d] Událost: event“ (bez uvozovek, kde „d“ je alias „d“ a „event“ je proměnná event) a celý řetězec přiřad'te na konec souboru xxx.log (který je v aktuálním adresáři).**

Alias d='date'; echo "[ `d ` ] Událost: \$event" >> xxx.log

**Vypíšte počet souborů končící na “.c“ nebo “.cpp“ které jsou umístěny v rodičovském adresáři domovského adresáře.**

Ls -l ~/./\*.{c,cpp} | wc -l

[https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos\\_test\\_2013-10-17-1](https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos_test_2013-10-17-1)

**Spust'te powerpoint z podadresáře local/bin s argumentem \$\$**

./local/bin/powerpoint '\$\$'

**Uložte do proměnné rozdíl počtu řádek souborů /etc/hosts a /etc/hosts.old**

PROM=\$((`cat /etc/hosts | wc -l` - `cat /etc/hosts.old | wc -l`))

**Do proměnné FTW uložte počet minut, které uplynuly od poslední půlnoci. Obsah proměnné poté vypíšte.**

FTW=\$((10#`date +%H`\*60 + 10#`date +%M`)); echo "\$FTW" -moje řešení

**Zobrazte všechna jména souborů v aktuálním adresáři, která obsahují řetězec uložený v proměnné FRAGMENT\_JMENA. (Před i za tedy může být libovolný počet znaků.)**

ls -d \*"\$FRAGMENT\_JMENA"\*

**Založte proměnnou TEXT tak, aby zaručeně nebyla proměnnou prostředí.**

Unset TEXT; TEXT="text";

### Vytvorte adresarovou strukturu:

**fs – FS (/etc/default/fs... LOCAL =ufs)**

**- PREFIX -svc**

**- log**

**- sh**

`mkdir -p fs-`cat /etc/default/fs | awk -F'=' '{print $2}'` ;` ..tvorba adresaroveho systemu

`mkdir -p fs-ufs/"$PREFIX"-svc/{log,sh} ;`

**Do promenne LIST ulozte obsah vseh adresaru z promenne PATH. Napoveda: obsah promenne PATH je v tomto tvaru: “/usr/lib:/usr/local/bin:/usr/sbin:”**

`LIST=$(ls $(IFS=: ; echo $PATH))`

**Do promenne WIKI ulozte „en.wikipedia.org“. Prikazem wget stahnete „http://WIKI/wiki/Shell“, kde WIKI je obsah promenne. Web zadejte prikazu wget jako parametr oddeleny mezerou. Potlacte vypis a chybove hlasi.**

`WIKI=en.wikipedia.org`

`wget http://$WIKI/wiki/Shell 1>/dev/null 2>&1`

**Spočítejte všechny soubory s příponou .c nebo .cpp ve složce data.**

`ls ~/data/*.{c,cpp} | wc -w`

**Vytvořte proměnnou A a uložte do ní hodnotu 5. Poté vypíšte hodnotu součtu této proměnné s proměnou UID a tu uložte do souboru test.txt**

`A="5"`

`echo $((A+$UID)) >> test.txt`

### Poznamky:

Vzdalene pripojeni + prikaz, pomoci promenne: `ssh "$USER"@fray1.fit.cvut.cz finger`

Vytvoreni souboru pomoci slozenych zavorek: `touch {s1,s2}{1,2,3}` , `touch {p,}{1,2}` , `touch s{1,2,3}{.txt,}`

' ' jednoduche uvozovky, "" slozene/dvojite uvozovky

"" : nebere \$, \, ``

' ' : v tom jsou vsechny znaky bezne

& : da neco na pozadi

&& : pokud prikaz vykonan kladne, pak se vykona dalsi prikaz

|| pokud se provede zaporne provede se

` ` identicke k \$( )

\$0 ..jmeno vykonavajiciho se programu

oddelovac..promenna IFS

`ls *q` ..nevylistuje soubory zacinajici teckou..nutne `ls .*q`

`ls ?` ..jeden znak

`ls [^a-z]` ..^ negace, ne zacatek! ..stejne jako !

Whereis ..prikaz, ukaze plnou cestu

jak zachovat formatovani prikazu: `echo "$cmd"` ..formatovane, `echo `cmd`` ..bez formatovani

`cat /etc/resolv.conf` .. zmena domain fit.cvut.cz ..nefunfuje :O

`ssh fray1.fit.cvut.cz "ls -a; /usr/ucb/whoami"` ..nutnost vypsati plnou cestu

`ssh fray1.fit.cvut.cz "ls -a; echo \"$USER\";"` ..? ..pwd-`pwd`

`echo "xxx${U}xxx"` ..xxxhanovgabxxx

`ssh "$U@fray1.fit.cvut.cz" 'ls -a; echo "username=$USER, pwd=$PWD"'`

`ssh "$U@fray1.fit.cvut.cz" 'ls -a; echo "username=\$USER, pwd=\$PWD"'`

`ssh "$U@fray1.fit.cvut.cz" 'echo "username=\"$USER\", pwd=\"$PWD\""'` ..vypisuje:

`username='hanovgab', pwd='/home/stud/hanovgab'`

`ls -ltr | tail -1` ...posledni modifikovany soubor, adresar

`scp /etc/resolv.conf "$U@$S:balbal"; ssh "$U@$S" 'ls -ltr | tail -2'`

^C+A ..skoci na zacatek

`ssh "$U@$S" "mkdir \"$NAME\""; scp /etc/resolv.conf "$U@$S:$NAME"; ssh "$U@$S" "ls -ltr \"$NAME\" | tail -2"`

pokud bychom kopirovali cely adresar, prepinač -r, rekurzivne ..pokud nevime, radeji s r!!!

`ssh "$U@$S" "mkdir \"$NAME\""; scp -r /etc/rc* "$U@$S:$NAME"; ssh "$U@$S" "ls -ltr \"$NAME\" | tail -2"`

`scp -r "$U@$S:/etc/rc*" "$HOME/$NAME"; ls -ltr "$NAME" | tail -2`

kdyz vyhodnoceni na vzdalene strane : rc\* chránit, na lokalu – chráněné být nesmí!

- Potlacte vypis a chybove hlaseni: **1>/dev/null 2>&1**

**set** – vypise lokální i globální proměnné

**env / printenv** - pouze globální proměnné

PAGER=less man man ..změní pouze jeden příklad, PAGER=less ; man man ; ..změněno v aktuálním shellu

změna promptu: PS1=\\d, PS1='\\d' ..**osetřit lomítko**

změna promptu pomocí aliasu: alias ps1='PS1=\\d' ..ne příkazové úvozovky!

touch {...,aa,b,XX}..{Z,O} ..vytvoří soubory typu aa.O aa.Z (plus skryté soubory)

**hvezdicka skryté soubory neexpanduje**

echo [a-z]\* ..vypíše neformatované soubory

prepínací u **echo -e** : lze použít speciální znaky, pr. \\n

cd ~cermada1/.. ..nadrážený adresář (libovolného) domovského

Úvozovka " , apostrof ' '

alias d='date "+%S %Y"'; echo `d` >> ~/.bashrc ..spatně ne?!

echo "alias d='date\\\\"+"%S %Y\*\\\""

echo "alias d='date \\\"+\\\$param\\\""; alias d='date \\\"+\\\$param\\\""

nastavení **lokální proměnné**: unset P; P=hodnota

printenv P; echo \$?; ..pokud nalezena 0, nenalezena 1

A="\$PS1"; printenv PS1; unset PS1; PS1="\$A";

**pro další přihlášení: uložit do ~/.bashrc**

echo "PS1='\$PS1'" >> ~/.bashrc

příkaz locale = nastavení prostředí

Při scp musí být udána slozka, kam se mají soubory uložit!! (zde ~)

scp -r ./"\$PREFIX"\* hanovgab@fray1.fit.cvut.cz:~

## **PRIPRAVA K 2. TESTU**

[https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos\\_test\\_2013-11-4-1](https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos_test_2013-11-4-1)

**Máme script ProgTest, který pro vstup používá stdin a pro výstup stdout. V nadřazeném adresáři jsou vstupní data v souboru 0000\_in.txt a výstupní data 0000\_out.txt. Jak zjistíme, že script funguje správně.**

./Progtest < ../0000\_in.txt | diff - ../0000\_out.txt - cte ze stdin

**Máme výpis ldd /bin/ls (byla ukáзка) a všechny soubory zde uvedené je potřeba vypsát jako ls -l a jako oddělovač pouze jedna mezera.**

ls -l \$(ldd /bin/ls | cut -d' ' -f2 | sed 's/^\\/' | tr -s '\\n' ' ' ..minus za f2: až do konce

nebo cut -d'>' -f2-

**N je proměnná ve které je uložena celočíselná hodnota X. Výstup následujícího uložte do souboru Xth. Máme soubory s1 a s2 z těchto souborů vezměte X-tý řádek od konce a uložte ho do souboru Xth.**

Cat s1 | tail -"\$N" | head -1 > "\${X}th"; cat s2 | tail -"\$N" | head -1 >> "\${X}th"

[https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos\\_test\\_2013-11-5-1](https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos_test_2013-11-5-1)

**Máte program sic pro ProgTest. V nadřazeném adresáři se nachází soubor s daty pro vstup 0000\_in.txt a pro výstup 0000\_out.txt. Zjistěte, jestli program funguje správně.**

./sic < ../0000\_in.txt | diff - ../0000\_out.txt

**Na první řádek souboru S přidejte záznam o jeho poslední modifikaci (výpis z ls -l S) ve formátu [datum=datum] (například [datum=Nov 5 18:30]).**

cat S > pom;

ls -l S | awk '{printf("[datum=%s %d %s]\\n", \$6,\$7,\$8)}' > s1;

cat pom >> s1

**V domovském adresáři se nachází soubor config. Načtěte do proměnné F hodnoty všech specifikovaných proměnných. Vypište na výstup. Soubor config může obsahovat cokoliv. Config:**

```
local_dir=~/project for =everyone=
```

```
filename=main.java
```

```
remote_host=dray1.feld.cvut.cz
```

```
F=`cat ~/config | cut -d'=' -f2-`; echo "$F" ..pokud by byl radek, kde se neprisazuje, sezere taky
```

**Z výstupu ypcat passwd zjistěte nejčastější křestní jméno. Poznámka: formát byl jako na frayi - křestní vždy první a oddělené mezerou.**

```
ypcat passwd | cut -d':' -f5 | awk '{print $1}' | grep -v 'testovací' | sort | uniq -c | sort -k1,1nr | head -1 | awk '{print $2}'
```

**Smažte složky obsažené v souboru data.txt a na výstup vypište počet neúspěšných mazání.**

```
Rm -r $(cat data.txt) 2>&1 | wc -l ..bez 1>/dev/null pac rm nevypisuje stdout
```

**Na výstup vypište název podadresare, která obsahuje nejvíce podadresářů.**

```
ls -l | grep '^d' | awk '{print $2}' | sort -k1,1nr | head -1 | awk '{print $2}'
```

rozdíl NF a \$NF (vypíše poslední položku)

```
ls -l | grep '^d' | tr -s ' ' | cut -d' ' -f2,9- | sort -k1,1nr | head -1 | cut -d' ' -f2- ..osečuje mezery v názvu
```

**Do proměnné NECO uložte nejvyšší UID z /etc/passwd.**

```
NECO=$(cat /etc/passwd | cut -d: -f3 | sort -k1,1nr | head -1)
```

**Máte zdroják kuku.c, který neprošel Progtestem a soubor oprava.neco, který obsahuje opravu. Opravte původní kuku.c. Napište jak vzniknul soubor oprava.neco.**

```
patch kuku.c < oprava.neco .
```

#soubor oprava vznikl takto:

```
diff -u kuku.c oprava > oprava.neco
```

#pro vygenerování obsahu souboru pro patch musí mít diff prepínac -u !!!!

**Z aktuálního adresáře vypište 5 největších obyčejných souborů (ideálně pouze jméno) končících na .txt . Výstup uložte do souboru max a zároveň jej přiřipšte za konec souboru /tmp/maxfiles.**

```
ls -l *.txt | grep '^-' | tr -s ' ' | cut -d' ' -f5,9- | sort -k1,1nr | cut -d' ' -f2 | head -5 | tee max >> max2
```

```
du *.txt | sort -k1,1nr | head -5 | cut -f2- | tee max >> max2
```

[https://www.fit-wiki.cz/%C5%A1kola/p/%C5%99edm%C4%9Bty/bi-uos/uos\\_test\\_2013-11-5-2?&#%C5%99e%C5%A1en%C3%AD](https://www.fit-wiki.cz/%C5%A1kola/p/%C5%99edm%C4%9Bty/bi-uos/uos_test_2013-11-5-2?&#%C5%99e%C5%A1en%C3%AD)

**Vypište nejčtenější příjmení z ypcat passwd. Pokud jich je více se stejnou četností, tak zobrazte který chcete.**

```
ypcat passwd | cut -d':' -f5 | cut -d' ' -f2 | grep -v 'student' | grep -v 'konto' | sort | uniq -c | sort -k1,1nr | head -1 | awk '{print $2}'
```

**Uložte nejmenší UID z příkazu ypcat passwd do proměnné MINU.**

```
MINU=`ypcat passwd | cut -d":" -f3 | sort -k1,1n | head -1`
```

```
echo "$PWD" | awk -F/ '{if(NF>2) {printf("%s\n",$(NF-1))} else print "/"}' ..vypsání rodičovského adresáře
```

```
file ~/dir/* | grep "directory$" | cut -d":" -f1
```

```
test -d "a" && echo ano, a je adr
```

```
du ..bloky, du -k ..kila
```

[https://www.fit-wiki.cz/%C5%A1kola/p/%C5%99edm%C4%9Bty/bi-uos/uos\\_test\\_2012-10-29](https://www.fit-wiki.cz/%C5%A1kola/p/%C5%99edm%C4%9Bty/bi-uos/uos_test_2012-10-29)

**Napište příkaz (příkazy), který vypíše jména uživatelů, kteří jsou uvedeni v souboru /etc/group, ale nejsou uvedeni v souboru /etc/passwd. Každé takové jméno necht' je vypsáno pouze jednou. Prázdné řádky ve výstupu nemají být. (Tip: Jména: 4. položka souboru /etc/group, jména jsou oddělena čárkou; 1. položka souboru /etc/passwd)**

```
cat /etc/group | cut -d: -f4 | sed '/^$/d' | tr , '\n' | sort | uniq > group
```

```
cat /etc/passwd | cut -d: -f1 | sort > pass
```

```
comm -23 group pass
```



comm: -1: Suppresses the output column of lines unique to file1.  
-2: Suppresses the output column of lines unique to file2.  
-3: Suppresses the output column of lines duplicated in file1 and file2.

**Napište příkaz, který vypíše podrobné informace o domovských adresářích všech uživatelů, seřazené dle jmen těchto adresářů. Pokud by se vyskytlo více stejných domovských adresářů, informace se vypíše jen jednou. Zajistěte, aby se reakce na neexistující adresář nevypisovala. (Tip: 6. položka souboru ypcat passwd. Nevypisujte obsah adresáře, ale jen informace o něm samém.)**

```
ls -ld `cat /etc/passwd | cut -d':' -f6 | grep -v '^$' | sort | uniq` 2>/dev/null
```

[https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos\\_test\\_2011-11-01](https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos_test_2011-11-01)

**Vytvořte alias uzivatele, který ze souboru /etc/passwd vybere uživatele, kteří mají shell /bin/false/ a do jedné řádky vypíše seznam jejich uživatelských jmen, oddělených čárkami.**

```
cat /etc.passwd | grep '/bin/false$' | cut -d':' -f1 | awk 'BEGIN{x=0;}{if(!x)printf("%s", $1); else printf(",%s", $1); x+=1;}' | tr '\n' ' '
```

**Ze souboru /usr/dict/words vezměte čtyřpísmenná slova nezačínající samohláskou, seřadte je reverzně a pošlete na chybový výstup.**

```
Grep '<\....\>' /usr/dict/words | grep -v '^[aeiou]' | sort -r | 1>&2
```

u ls -l nahore radek total... odstranit

grep..zacatek, konec slova: \< \>

wc -l soubor (vypise jmeno souboru) x wc -l < soubor (vypise pouze cislo)

## Textove filtry I.

1. Vypíšte podrobnosti o všech login shellech všech uživatelů (o každém shellu právě jednou) [7.položka souboru/etc/passwd] : **ls -l `cat /etc/passwd | cut -d: -f7 | sort -u`**
2. V souborech s1, s2 a s3 jsou zvlášť informace o názvu souborů, jejich velikosti a vlastníkov. Spojte tyto soubory v uvedeném pořadí a seřadte výstup podle vlastníka : **paste s1 s2 s3 | sort -k3**
3. Máte k dispozici soubor s1 a soubor p, který obsahuje změny souboru s2 oproti souboru s1. Použijte vhodný příkaz, který provede změny v souboru s1 aby odpovídal souboru s2. Jak byste takový soubor p získali? : **(diff -u s1 s2 > p;) patch s1 < p**
4. V souboru shells.denied jsou uvedeny zakázané login shelly. Zjistěte, které z nich jsou uvedeny v souboru /etc/passwd – (7.položka) : **cat /etc/passwd | cut -d:' -f7 | sort -u | grep `cat shells.denied`**
5. Kolik různých login shellů se používá (v souboru /etc/passwd – 7.položka)? : **cat /etc/passwd | cut -d:' -f7 | grep -v '^\$' | sort | uniq | wc -l**
6. V souboru s1 jsou uvedeny názvy adresářů. Vytvořte tyto adresáře a spočítejte, kolik z nich nemohlo být vytvořeno (např. protože již existovaly). : **mkdir `cat s1` 2>&1 | wc -l** (nezkontrolováno)
7. V domovském adresáři je soubor config s následujícím obsahem (řádky mají libovolné pořadí) Do proměnných local\_dir, filename, remote\_host, remote\_dir vložte hodnoty z config souboru (stačí provést vzorově pro jednu proměnnou). : **FILENAME=~ /cat config | grep '^filename=' | cut -d= -f2`** (nezkontrolováno)
8. V adresáři local\_dir jsou podadresáře, jejichž název označuje verzi projektu ve formátu main.major.minor (např 2.6.17). Do proměnné cur vložte označení poslední verze a do proměnné prev vložte označení předposlední verze projektu. Vytvořte soubor p obsahující změny mezi verzemi prev a cur. Soubor p přeneste na vzdálený počítač remote\_host do adresáře remote\_dir. : **cur=`ls local\_dir/ | sort -nr | head -1` ; prev=`ls local\_dir/ | sort -nr | head -2 | tail -1` ; diff -ur "\$prev" "\$cur" > p; scp p "\$USER"@remote\_host:remote\_dir;**

9. V proměnné RC má být uloženo rodné číslo ve formátu nnnnnn/nnn případně nnnnnn/nnnn. Ověřte pomocí příkazu grep, že obsah proměnné RC odpovídá formátu (n=cifra). : **echo "\$RC" | grep -c '^([0-9]{6})/([0-9]{3,4})\$'** ..grep -c ... vypise pocet radek odpovidajich zadani
10. Vypište 5. řádky ze souborů s1 a s2 do souboru /tmp/x-username (kde username je uživatelské jméno). : **cat s1 | head -5 | tail -1 > /tmp/x-username; cat s2 | head -5 | tail -1 >> /tmp/x-username;**
11. Do proměnné MAX uložte největší trojčíferný UID (3. sloupec oddělený „:“) z výstupu ypcat passwd. : **MAX=ypcat passwd | cut -d ":" -f3 | grep '^([0-9]{3})\$' | sort -nr | head -1`**
12. Zkopírujte soubor /home/courses/BIUOS/public/pool/files/apache.log do aktuálního adresáře. Chyby ze souboru apache.log, tj. řádky obsahující [error], přidejte na konec souboru error.log.

Oct 12 08:12:07 edux apache2[rxp]: [Mon Oct 12 08:12:07 2009] [error] [client 14 7.32.31.193] request failed: in reading headers.

**Cat apache.log | grep '[error\]' >> error.log** ...[ ] vypise radky obsahujici zadane znaky

13. Ze souboru apache.log vypište všechny různé IP adresy seřazené podle počtu požadavků GET z těchto adres. : **cat apache.log | grep '"GET /' | cut -d " " -f6 | sort | uniq -c | sort -nr | awk '{print \$2}'**
14. Smažte všechny soubory z aktuálního adresáře, které obsahují (uvnitř) řádku #!/bin/bash nebo #!/bin/ksh. : **rm `egrep -l '#!/bin/bash|#!/bin/ksh' .\*`**
15. Kolik místa na disku zabírají dynamicky linkované knihovny programu /usr/bin/ls? (Seznam knihoven najdeme například ve výstupu příkazu ldd /usr/bin/ls.) : **ls -l `ldd /usr/bin/ls` | cut -d ">" -f2` | awk 'BEGIN{x=0;}{x+=\$5;}END{printf("%d\n",x);}'**
16. Vytvořte symbolický link s názvem symlink, který odkazuje na soubor uvedený na 3. řádku proměnné LINK\_TO (proměnná má tedy více řádků!). : **ln -s `echo "\$LINK\_TO" | head -3 | tail -1` symlink** (zkontrolovat)
17. Do proměnné GMAX uložte největší čtyřčíferný GID z výstupu ypcat passwd nebo z výstupu ypcat group (passwd: 4. sloupec oddělený „:“, cagroup: 3. sloupec oddělený „:“). : **GMAX=ypcat passwd | awk -F: '{print \$4}' | grep '^([0-9]{4})\$' | sort -nr | head -1`** ..nebo group?? ..nezapominat ^ a \$ u grep!
18. Ze souboru apache.log odstraňte chyby, tj. řádky obsahující [error], a spočítejte požadavky GET na soubory končící na jpg. : **cat apache.log | grep -v '[error\]' | grep '"GET.\*\jpg' ..rm?**
19. Ze souboru apache.log vypište všechny různé HTTP metody (1. slovo v uvozovkách) seřazené podle počtu jejich výskytů. : **cat apache.log | grep -v '[error\]' | cut -d "'" -f2 | awk '{print \$1}' | grep -v '^\$' | sort | uniq -c | sort -k1,1n | awk '{print \$2}'** (zkontrolovat)
20. Přesuňte všechny soubory z aktuálního adresáře, které obsahují (uvnitř) řádku #!/bin/bash, do adresáře scripts. Nezapomente na testovací soubory, napr. Cp /etc/init.d/de\* **mv `egrep -l '#!/bin/bash' .\*` script/**
21. Máte soubory s1, s1.head a s1.mid. V souboru s1.head jsou údajně řádky 1-10 (včetně) a v s1.mid jsou údajně řádky 30-40 (včetně) ze souboru s1. Zjistěte, zda je to pravda. : **diff <(cat s1|head) s1.head && diff <(cat s1|head -40 | tail -11) s1.mid** ???
22. Z výpisu ypcat passwd vypište pouze uživatelská jména a login shell uživatelů, jejichž uživatelské jméno začíná na novak. Jméno a login shell oddělte znakem |. Na konci seznamu napište číslo, které představuje počet takových uživatelů. : **ypcat passwd | grep '^novak' | sort -k1,1 | uniq | awk -F: 'BEGIN{x=0;}{print \$1|"|\$7; x+=1;}END{print x;}'**
23. Zjistěte, zda soubor user.max\_uid skutečně obsahuje jméno uživatele s maximálním UID (z výpisu ypcat passwd) ...nalezeni max uid | **grep -c `cat user.max\_uid`**
24. Z výpisu ypcat passwd vypište pouze UID a GID uživatelů, jejichž uživatelské jméno začíná na novak. UID a GID oddělte znakem @. Na konci seznamu napište číslo, které představuje počet různých skupin takových uživatelů. : **ypcat passwd | grep '^novak' | sort -t: -k4,4n -u | awk -F: 'BEGIN{x=0;}{printf("%d @ %d\n", \$3, \$4); x++;}END{print x}'**
25. Zjistěte, zda soubor user.max\_gid skutečně obsahuje jméno uživatele s maximálním GID (z výpisu ypcat passwd). : ...nalezeni max gid | **grep -c `cat user.max\_gid`**
26. Z výpisu ypcat passwd vypište pouze username a UID uživatelů, jejichž GID je 1001 a jejich username začíná na novak. Username a UID oddělte znakem \*. : **ypcat passwd | grep '^novak' | uniq | awk -F:**



'{if(\$4 == 1001){print \$1 "\*" \$3}}'

zkontrolovat! Plus sort -k1

27. Z konfiguračního souboru /etc/sshd.conf vypište poslední výskyt direktivy (řetězce) PermitRootLogin, který není zakomentovaný (tj. před řetězcem se nenachází znak# ). : **cat PermitRootLogin | grep -v '#.\*PermitRootLogin' | grep 'PermitRootLogin' | tail -1**

**cat PermitRootLogin | grep '^[^#]\*PermitRootLogin' | tail -1;**

28. Který uživatel (dle /etc/group) má přiřazeno nejvíce skupin?

**cat /etc/group | cut -d: -f4 | grep -v '^\$' | tr ',' '\n' | sort | uniq -c | sort -k1,1n | tail -1**

29. Kteří uživatelé uvedení v /etc/group nejsou definováni v lokálním seznamu uživatelů?

30. Ve kterých skupinách jsou tito neexistující uživatelé?

31. Ze souboru biuos-c06-apache.log v adresáři /home/courses/BIUOS/public/ vytvořte statistiku stahování metodou GET pro soubory s příponou .html. Statistiku zobrazte formou tabulky - pro každý soubor vypište řádek: „počet stažení, jméno souboru“ Zjistěte, který soubor byl nejčastěji stahovaný a kolikrát.

32. Vypište počet souborů, které se nalézají na všech cestách specifikovaných proměnnou PATH (bez duplicit), seříděný seznam uložte do souboru commands.

33. Kolik různých slov skládajících se z pěti písmen obsahuje soubor /usr/dict/words ? : **cat /usr/dict/words | sed -n '/^\.{5}\$/' | sort -u | wc -l** ..words obsahuje pouze písmena, ne jiné znaky

34. Kolik skriptů pro shell “/bin/bash“ vykonatelných ostatními uživateli je v daném adresáři a jeho podadresářích? Jméno adresáře je uloženo v souboru souboru DIR. Zobrazte jména těchto skriptů a jejich celkový počet.

35. Kolik různých slov skládajících se z pěti písmen malé abecedy obsahuje výstup příkazu man man ? : **man man | tr " " "\n" | grep '^[a-z]\{5}\$' | sort -u | wc -l** ..nevezme pr. Pokus.

Man man | tr ' ' .-[ '\n\*' | ... doladit

36. Ze souboru /etc/services spočítejte, kolik různých služeb je registrováno na transportním protokolu tcp. Zobrazte seznam jejich jmen s čísly portu.

37. Kteří uživatelé se přihlásili nebo odhlásili k serveru fray1 ve srovnání se stavem před minutou? (Vícenásobné připojení počítáme jako jedno.)

### PRIPRAVA K 3. TESTU

[https://www.fit-wiki.cz/%C5%A1kola/p/%C5%99edm/%C4%9Bty/bi-uos/uos\\_test\\_2013-11-25-1](https://www.fit-wiki.cz/%C5%A1kola/p/%C5%99edm/%C4%9Bty/bi-uos/uos_test_2013-11-25-1)

**Příkazem who vypište seznam uživatelů, seřad'te podle abecedy, pouze unikátní jména. Ve výpisu za jméno spočítejte, kolik každé jméno obsahuje znaků. Očekávaný výstup:**

**machvojt (8)**

**root (4)**

**safondmy (8)**

**volcijak (8)**

- 1) **who | sort -k1,1 -u | awk '{printf("%s (%d)\n",\$1,length(\$1))}'**
- 2) **who | cut -d' ' -f1 | sort -u | awk '{print \$0 ("length(\$0))"}'**

**Zkopírujte všechny soubory z aktuálního adresáře a podadresářů, které končí na .c, .h, .sh nebo .m, starších než týden, s nenulovou velikostí do adresáře uloženého v proměnné DIR (proměnná DIR může obsahovat bílé znaky).**

**find . -type f -ctime +7 -size +0c \( -name '\*.c' -o -name '\*.h' -o -name '\*.sh' -o -name '\*.m' \) -exec cp {} "\$DIR" \;**

**Do proměnné UMASK uložte aktuální masku. Nastavte masku na hodnotu 27. Vytvořte adresář test v aktuálním adresáři. Zjistěte nastavení práv nového adresáře test a práva uložte jako text (např. rw-rw-r-) do proměnné PERM. Nastavte masku na původní hodnotu z proměnné UMASK.**

**UMASK="`umask`"; umask 27; mkdir test; PERM="`ls -ld test | cut -d' ' -f1 | tr -d d`"; umask "\$UMASK"**

**nebo PERM="`ls -ld test2 | tr "d" " " | cut -d' ' -f2`"**

(tr -s ' ' ...nahradí více mezer pouze jednou / smazání všech mezer v textu: sed 's/[ ]//g' ) )

**Zkopírujte všechny soubory a soubory v adresářích uvedených v souboru test.txt, které mají uvedenou absolutní cestu (některé soubory a adresáře jsou uvedeny s absolutní cestou jiné nejsou) do adresáře uvedeném v proměnné ADR (proměnná ADR může obsahovat bílé znaky).**

/etc/passwd

history.bc

./home/

main.c

/bin

find `awk '{print \$1}' test.txt | grep '^/.\*' -type f -exec cp {} "\$ADR" \;

resení prebytečných mezer na začátku řádku: awk '{print \$1}' (nelze sed 's/[ ]//g' pac odstraní mezery v názvech souboru)

**Najděte soubory kterých jste vlastníkem a změňte jejich práva na: vlastník vše, skupina číst, ostatní nic.**

find -type f -user "\$USER" 2>/dev/null -exec chmod 740 {} \; ..lze nezadat cestu

- egrep <flags> '<regular expression>' <filename>

- Some common flags are: **-c** for counting the number of successful matches and not printing the actual matches, **-i** to make the search case insensitive, **-n** to print the line number before each match printout, **-v** to take the complement of the regular expression (i.e. return the lines which *don't* match), and **-l** to print the filenames of files with lines which match the expression.

**Všechny soubory z domovského adresáře s koncovkou .sh nebo .ksh obsahující řádky #!/bin/bash nebo #!/bin/ksh zkopírujte do složky scripts v domovském adresáři. Chyby zahodte.**

egrep -l '#!\bin\bash|#!/\bin\ksh' ~/\*.sh ~/\*.ksh 2> /dev/null | **xargs -I {}** cp {} ~/script 2> /dev/nul

find ~ -type f \( -name "\*.sh" -o -name "\*.ksh" \) -exec egrep -l '#!\bin\bash|#!/\bin\ksh' {} \; -exec cp -f {} ~/scripts/ \; 2>/dev/null

**find** \ (mezera ... mezera\)

**Vypište S: (počet souborů POUZE v aktuálním adresáři) V: (celková velikost souborů) B.**

gfind . -maxdepth 1 -type f -exec du {} \; | awk 'BEGIN { x=0; } { x+=\$1; } END { print "S: " NR " V: " x " B" } '

rozdíl mezi gfind a find: ve find nefunguje maxdepth

NR : počet řádku

awk /vzor/ { akce } pr. awk /HG/ par.log ..nalezne řádek obsahující HG

BEGIN { akce } : Provede příkazy akce na začátku běhu skriptu ještě před tím, než jsou zpracována vstupní data.

END { akce } : Podobně jako v předchozím případě, ale akce se provede až na konci běhu skriptu.

**Najděte ve složce a podsložkách Program Files soubory které neobsahují v názvu mezeru a vypište je.**

Nejspíš nejdřív najít findem Program Files, pak na to udělat něco jako ..find (?) -type f \! -name "\*" "\*" 2>/dev/null

[https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos\\_test\\_2013-11-26-1](https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%9Bty/bi-uos/uos_test_2013-11-26-1)

**V aktuálním adresáři a případných podadresářích najdete všechny soubory, pro které má vlastník právo alespoň na čtení a vypište o nich všechny dostupné informace.**

find . -type f -perm -u=r -ls (nebo -perm -400)

**V aktuálním adresáři se nacházejí podadresáře. Soubory z podadresářů (pouze podadresářů), jejichž jméno (myšleno jméno podadresáře) začíná na dvě dekadické číslice a je následováno podtržítkem (např. Adrsář 12\_UOS) obsahující na začátku některé řádky řetězce [err]: nebo [asdf]: odstraňte. Chybová hlášení zahod'te.**

egrep -l '^\[err\]:|^\[asdf\]:' ./[0-9][0-9]\_\*/ 2>/dev/null | xargs -I {} rm {}

**V aktuálním adresáři existuje soubor f.txt, který obsahuje seznam souborů (máte garantováno, že bez bílých znaků). Nastavte těmto souborům práva tak, aby je mohli číst a jen číst všichni uživatelé.**

Chmod a=r `cat ./f.txt`

**Z výpisu ypcat passwd uložte informace o aktuálním uživateli do souboru konto v domovském adresáři. Pokud je shell /bin/bash nahradte ho za /bin/csh**

1) ypcat passwd | grep "^\${USER}:" | sed 's:/bin/bash\$/:/bin/csh/g' > ~/konto

2) ypcat passwd | grep "^\${USER}:" | awk -F: '{print \$1":"\$2":"\$3":"\$4":"\$5":"\$6":"\$7"; if(\$7=="bin/bash"){print "bin/csh"}}'

**Najděte v domovském adresáři soubory které jsou spustitelné aktuálním uživatelem a změňte jejich práva, aby odpovídala -x-w-wx.**

gfind ~ -maxdepth 1 -type f -perm -100 -exec chmod 123 {} \;

**Připojte prvních 10 řádek ze souboru /neco/neco na začátek souboru pokus. Soubor pokus již existuje.**

cat pokus > tmp ; head -10 /neco/neco > pokus ; cat tmp >> pokus ; rm tmp

**V adresáři, jehož cesta je uložena v \$ROOT, vytvořte nový adresář, jehož jméno je uloženo v \$DIR. Pak zkopírujte všechny soubory z aktuálního adresáře které začínají na číslici do adresáře \$DIR. Proměnné mohou obsahovat prázdné znaky.**

mkdir -p "\$ROOT/\$DIR"; gfind . -maxdepth 1 -type f -name "[0-9]?" -exec cp {} "\$ROOT/\$DIR" \;

místo gfind: ls -d [0-9]\*

**V aktuálním adresáři najděte všechny soubory, jejichž jméno začíná na desítkovou číslici a na třetím místě je libovolné písmeno.**

ls -ld [0-9]?[\_]\* | awk '/^-/ {print \$9}'

[https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%Bty/bi-uos/uos\\_test\\_2013-11-28-1?&#%C5%99e%C5%A1en%C3%AD](https://www.fit-wiki.cz/%C5%A1kola/p%C5%99edm%C4%Bty/bi-uos/uos_test_2013-11-28-1?&#%C5%99e%C5%A1en%C3%AD)

**Přepište soubor /etc/group a ke všem GID kromě 0 přičtěte 1.**

- 1) cat /etc/group | awk -F: 'BEGIN{OFS=":"}{if(\$3){\$3++;} print \$0}' r > /etc/group (nejspíše třeba ještě osetřit práva pro zápis do /etc/group)
- 2) cat /etc/group | awk -F: '{OFS=":"}{if(\$3){\$3=\$3-1;}print \$0}' > etc; mv etc /etc/group;

**V aktuálním adresáři všem souborům, jejichž jméno končí na .sh nebo .bash přidejte právo na čtení s respektem k aktuální masce přístupových práv.**

chmod `echo \$((777 - 10#\$umask))` `ls \*.sh \*.c` 2>/dev/null

**V aktuálním adresáři a jeho podadresářích vyhledejte soubory, které aktuální vlastník (spouštějící příkaz) může alespoň číst.**

Find ~ -type f -user "\$USER" -perm -u=r

**Ve složce uložené v proměnné \$STARA máte fotky. Ve složce uložené v proměnné \$NOVA máte aktualizované tyto fotky a i spoustu dalších fotek. Zkopírujte pouze aktualizované verze z \$NOVA do \$STARA.**

cp "\$NOVA"/`ls "\$STARA"` "\$STARA"

## Prace s filtry

1. Vypište seznam procesů seřazený podle spotřeby procesorového času.
2. Vypište seznam procesů seřazený podle spotřeby paměti.
3. Vypište seznam procesů seřazený podle délky běhu.
4. Spusťte 5× po sobě příkaz p na pozadí (např. Sleep 1000)
  - Ze stejného terminálu ukončete naposledy spuštěný příkaz p.
  - Ze stejného terminálu ukončete třetí spuštěný příkaz p.
  - Ze stejného terminálu ukončete nejdříve spuštěný příkaz p.
  - Ze druhého terminálu ukončete svůj příkaz p, který běží nejdéle.
  - Ze druhého terminálu ukončete svůj příkaz p, který spotřeboval nejvíce procesorového času celkem.
  - Ze druhého terminálu ukončete svůj příkaz p, který nejvíce zatěžuje procesor.

- `grep "^\([A-Za-z][m-y]\).*\$1\$" words`
- `who | awk '{print $1}' | sort | uniq | awk '{print NR "=" $1}END{print "-----"; print "Celkem : " NR}'`
- kdyz chci vytvorit soubor v adresari – tyka se prav adresare
- `rw- rw- --- umask 660 (soubor), 007 (adresar)`
- `find -size, name, perm, user, mtime, links`
- `find ~ -type f \( -size 0c -o -mtime +7 \) -ls` nebo `find ~ -type f \( -size 0c -o -mtime +7 \) -exec ls -l {} \;`
- `find . -type f -user $USER -perm -u+r -ls`
- `find /home/courses/BIUOS -type f -size +0 \! \( -user root -o -user cermada1 \) | wc -l`
- Vypis nejdelsiho jmena: `ypcat passwd | cut -d ":" -f5 | awk '{print length($0), $0}' | sort -n | tail -1 | cut -d " " -f2-`
- `find . -type f \! -perm -u+r -ls`
- `find. -type f \( -name "*.chm" -o -name "*.sh" \) -size +0 -ls`
- `find. -type f \( -name "*.chm" -o -name "*.sh" \) -size +0 -mtime -6 -ls`
- `gtar xvzf cvic9/cvic9.tar.gz ?`
- `find. -type f \( -name "*.chm" -o -name "*.sh" \) -size +0 -ls | awk '{c+=$7}END{print c}'`
- `find. -type f \( -name "*.chm" -o -name "*.sh" \) -size +0 -exec du -k {} \; | awk '{c+=$1}END{print c*1024}'`
- ve findu.. rozdil shell. prom x regul. Vyrazy??
- `find x -type d -exec chmod 700 {} \;` ..adresare
- `find x \( -type f -exec chmod 600 {} \; \) -o \( -type d -exec chmod 700 {} \; \) 2>&1`
- relativni x absolutni archivace:
- relativni: `tar cvf neco?.tar x/*` vytvoreni
- `tar tvf 'neco?.tar'` ..pro zjisteni obsahu, t jako test
- `find . -mtime 1 ...presne cislo, presne 1 den stare`
- `find . -mtime -1 -exec tar cvf ../archiv.tar {} \+`
- ctime – vytvoreni, mtime – modifikace
- `find . -newer ../archiv.tar`
- `gzip archive.tar` ..smaze original; `gunzip archiv.tar.gz`; reseni: `presmerovani vstupu: gzip < archiv.tar > a.tgz`
- `time gzip < archiv.tar > a.tgz`
- `ls -lh` ..vypise velikost s jednotkami, pr. 1012K, 1,1M
- `prepinač p u taru` ..zachovani prav, o.. zachovani uzivatele
- `umask 022` : u souboru `rw- r-- r--` , u adresare `rw- r-x r-x` ..narozdil od adresare, u souboru neni implicitne povoleno spusteni x (rozdil mezi 666(soubor) / 777(adresar))
- pri vytvareni relativniho archivu .. treba prepnout dovnitr (cd) daneho adresare
- `pgrep -xl bash`
- vsichni uzivatele, kteri uzivaji bash
- `ps -p "`pgrep -x bash`" -o user= | sort -u`
- `ps -p "`pgrep -x bash`" -o user= -o ruser= -o uid= -o ruid="Toto je Real UID" | head` (= ..specifikuje hlavicku)
- `pgrep -x passwd`

### Prace se soubory, pristupovymi pravy

8) Nastavte masku pristupových prav tak, aby nově vytvořené soubory nebyly čitelné a zapisovatelné nikým, kromě vlastníka a skupiny vlastníka.

9) Do proměnné UMASK uložte aktuální hodnotu masky přístupových práv. Poté masku změňte na hodnotu 027 a vytvořte adresář test. Do proměnné PERM vložte textovou informaci o přístupových právech (např. `rw-r--r-x`) vytvořeného adresáře test. Nakonec nastavte masku na původní hodnotu podle proměnné PERM.

**UMASK=`umask`**; **umask 027**; **mkdir test**; **PERM=`ls -ld test | awk '{print \$1}'`** ; **umask "\$UMASK"**; ..plus jeste smazat d na zacatku.. neni pravo (sedem)

10) **Všem obyčejným souborům, které vlastníte, v domovském adresáři a jeho podadresářích, nastavte**

**přístupová práva odpovídající textovému zápisu rw-r-----.**

```
find ~ -type f -exec chmod 640 {} \;
```

11) Všem spustitelným souborům, jejichž jste vlastníkem, v domovském adresáři v adresáři skriptů a jeho podadresářích, nastavte přístupová práva čtení a spouštění pro všechny.

12) Všem obyčejným souborům v podadresáři projekty a jeho podadresářích, jejichž jméno končí na .c nebo .cp, nastavte taková přístupová práva, aby je mohl číst a zapisovat pouze vlastník souboru. Případné chyby zahod'te.

13) **Všem souborům (adresářovým položkám) v podadresáři POKUS domovského adresáře (jen v něm) nastavte přístupová práva odpovídající masce přístupových práv. Všechna práva tedy budou vypadat tak, jako by daná položka byla právě vytvořena příkazy touch, mkdir, nebo mknod. (Řešení je mnoho, ale některá jsou o řád obtížnější než jiná.)**

14) Všem souborům (adresářovým položkám), jejichž jméno končí na .sh nebo .bash přidejte právo pro čtení a spouštění tak, aby to respektovalo masku přístupových práv. Všimněte si, že na rozdíl od předchozího příkladu zde nemusíme rozlišovat adresář od ne-adresáře. Zdůvodněte si proč! (Není to koncovkami, na nich typ souboru nezáleží.)

15) **Všem podadresářům (rekurzivně) v domovském adresáři nastavte přístupová práva odpovídající masce přístupových práv. Na výstup vypište počet adresářů u nichž se změna povedla a počet adresářů u nichž se změna nepovedla ve formátu: povedlo / nepovedlo. Případné chyby zahod'te.**

```
povedlo=`gfind . -maxdepth 1 -type d -exec chmod $((777 - 10#$(umask))) {} \; 2>/dev/null | wc -l`;
nepovedlo=`gfind . -maxdepth 1 \! -perm -${((777 - 10#$(umask)))} 2>/dev/null | wc -l`;
echo "${povedlo} / ${nepovedlo}";
```

## ZAVERECNY TEST

V podadresari MM vytvorte symbolicky link jmenem DD.log, který ukazuje na první soubor (2. sloupec), který je uvedený direktivou (slovem) customlog a není zakomentovaný (nepředchází mu znak #) v souboru /etc/apache2/httpd-std.conf (DD je aktuální den v měsíci).

```
cat /etc/apache2/httpd-std.conf | grep -v '^#' | grep -i '^customlog' | awk '{print $2}'
```

Pokud link DD.log neukazuje na obyčejný soubor, připište do souboru test.log v adresari log radku: [HH:MM:SS] [error] soubor XXX neexistuje! (HH,MM,SS podle aktuálního času, XXX je cíl linku DD.log)

```
if [ -f `date '+%d`.log ]; then echo date "+[%H:%M:%S] [error] soubor $TARGET neexistuje!" >> test.log
```

```
"; fi
..předpokladám, že tento příklad dělám ve stejný den, kdy vytvarím log samozřejmě :D (jinak je třeba uložit název linku, pr. přes promennou)
```