LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

- პირველი კომპიუტერი
- ოპერაციული სისტემების განვითარება, Unix-ის ეპოქა
- ლიცენზიები
- პროგრამული უზრუნველყოფა
  - მესაკუთრეობრივი პროგრამები
  - საცდელი და უფასო პროგრამები (shareware/freeware)
  - თავისუფალი პროგრამული უზრუნველყოფა:
    - მოიცემა წყარო კოდთან ერთად, რომელშიც შესაძლებელია ცვლილების შეტანა.
    - შესაძლებელია მისი გავრცელება შეზღუდვის გარეშე.
    - შესაძლებელია მოდიფიცირებული კოდის გავრცელებაც.

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

პროგრამული უზრუნველყოფა	წვდომა წყარო კოდთან	ცვლილების შეტანა	გავრცელება	უფასო
მესაკუთრეობრივი	არა	არა	არა	არა
უფასო (freeware)	არა	არა	ყოველთვის არა	კი
საცდელი (shareware)	არა	არა	ყოველთვის არა	არა
თავისუფალი	კი	კი	კი	ყოველთვის არა

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

- **ყველასთვის ხელმისაწვდომი**
- **თანამშრომლობითი განვითარება**
  - ხარვეზების სწრაფად აღმოფქვრა
  - მრავალი პროგრამისტის ჩართულობა
- **ლინუქსის საზოგადოებები**
- **Unix-ის ფილოსოფია: « small is beautiful »**
- **სისტემის სტრუქტურა**
  - ბირთვი (kernel)
  - ბრძანებათა ხაზი
  - ბრძანებები

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

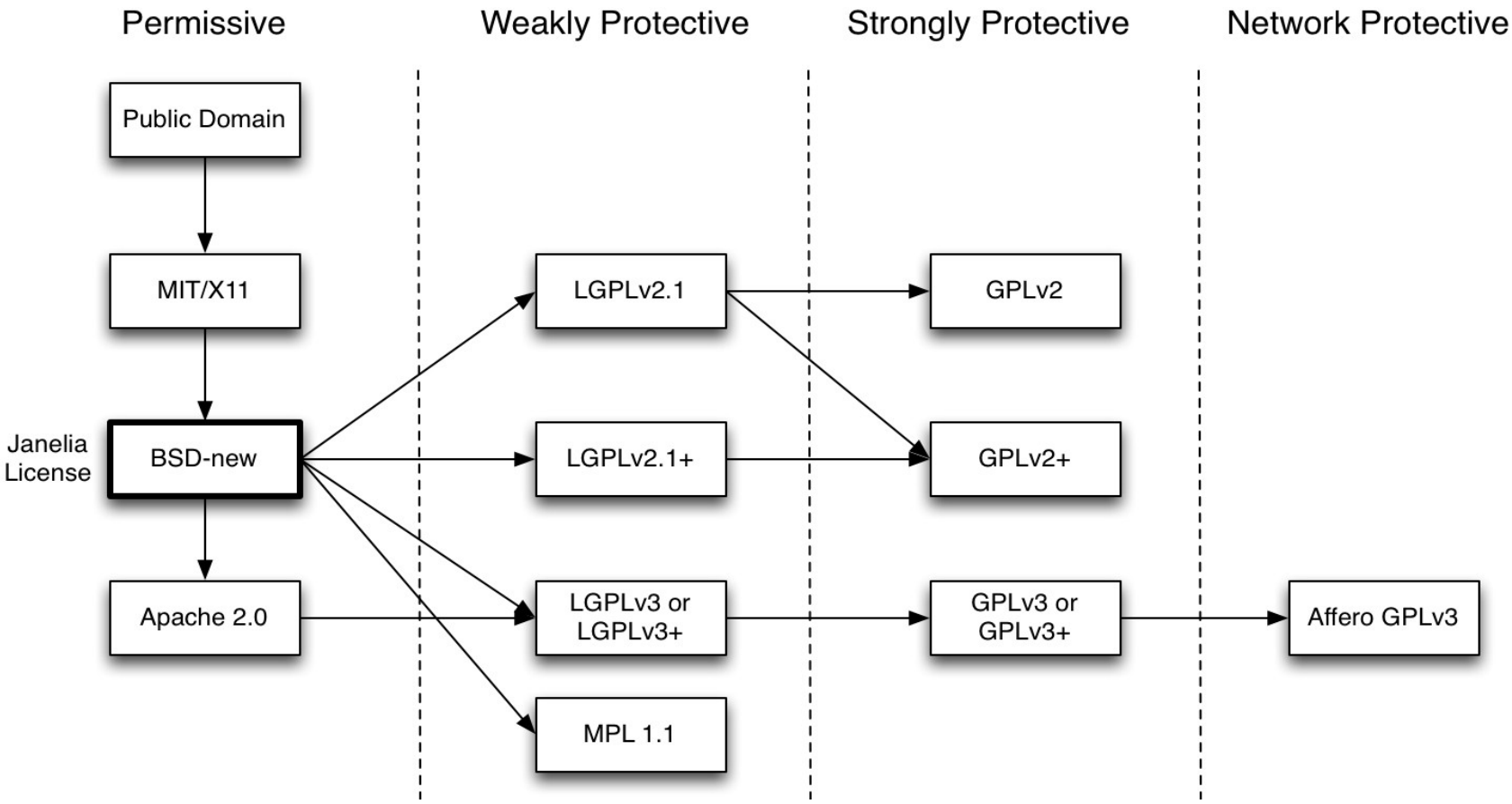
Ressource: https://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licenses

- 1983 წელს რიჩარდ სტალმანი (Richard Matthew Stallman) იწყებს პროექტს სახელად - GNU და მიზნად ისახავს შექმნას UNIX ტიპის ოპერაციული სისტემა, რომელიც მთლიანად თავისუფალი პროგრამებით იქნება დაკომპლექტებული.
- GCC (compiler/debugger GNU), ტექსტური რედაქტორი GNU Emacs ...
- 1985 წლის ოქტომბერში მან დააფუძნა Free Software Foundation (FSF).
- Copyleft ⓒ ; Copyright © ; Copyleft ლიცენზია მოითხოვს, რომ მის ქვეშ არსებული პროდუქტიდან შექმნილი/შეცვლილი პროდუქტი იმავე ლიცენზიით იქნას გამოსული.
- GPL (v1,v2,v3), Lesser/Library GPL (LGPL), GNU Free Documentation License (GFDL)
- არა copyleft ლიცენზიები (Apache, BSD, MIT/X11, Mozilla Public License ...)

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ლიცენზიები



LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

1990 წლიდან ...

- თითქმის დასრულებული GNU ოპერაციული სისტემა
- მიკრო ბირთვი - GNU Hurd არ არის ბოლომდე მიყვანილი
- მზარდი საზოგადოება
- აქტიური ფონდები უზრუნველყოფს იურიდიული ასპექტების მოგვარებას
- ლინუქსის (Linux) დასაწყისი :
  - ლინუს ტორვალდსი (Linus Torvalds) - PC 386, 4 MB RAM, 40MB HD
  - გამოვიდა 1991 წლის სექტემბერში; დეკემბერი, 1991 – v0.11; თებერვალი, 1992 – v0.12 (GPL-ის ქვეშ); მარტი, 1992 – v0.95; მარტი, 1994 – v1.0 (176 250 ხაზი კოდში)
  - მონოლითური ბირთვი
  - ჩატვირთვადი მოდულები

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

პირველი შეტყობინება

- 25/08/1991, comp.os.minix

> *Hello everybody out there using minix -*
>
> *I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).*
>
> …

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

- **ლინუქსის უპირატესობები:**
  - რეაქტიული
  - უსაფრთხო
  - მრავალამოცანიანი
  - მრავალმომხმარებლიანი
  - პორტატული: PC (i386 ... Pentium 4, AMD ...), itanium (IA 64), Sun (Sparc), Compaq Alpha, PowerPC (PowerMac, G3, G4, iMac, iBook, RS6000...), Mips (Colapt, SGI), architecture based on 68000 (Mac, Amiga), Arm (NetWinder), SuperH, Mainframe.
- **დისტრიბუტივები:** მარტივი, მძლავრი და მრავალენოვანი საინსტალაციო სისტემა.

LINUX სისტემები

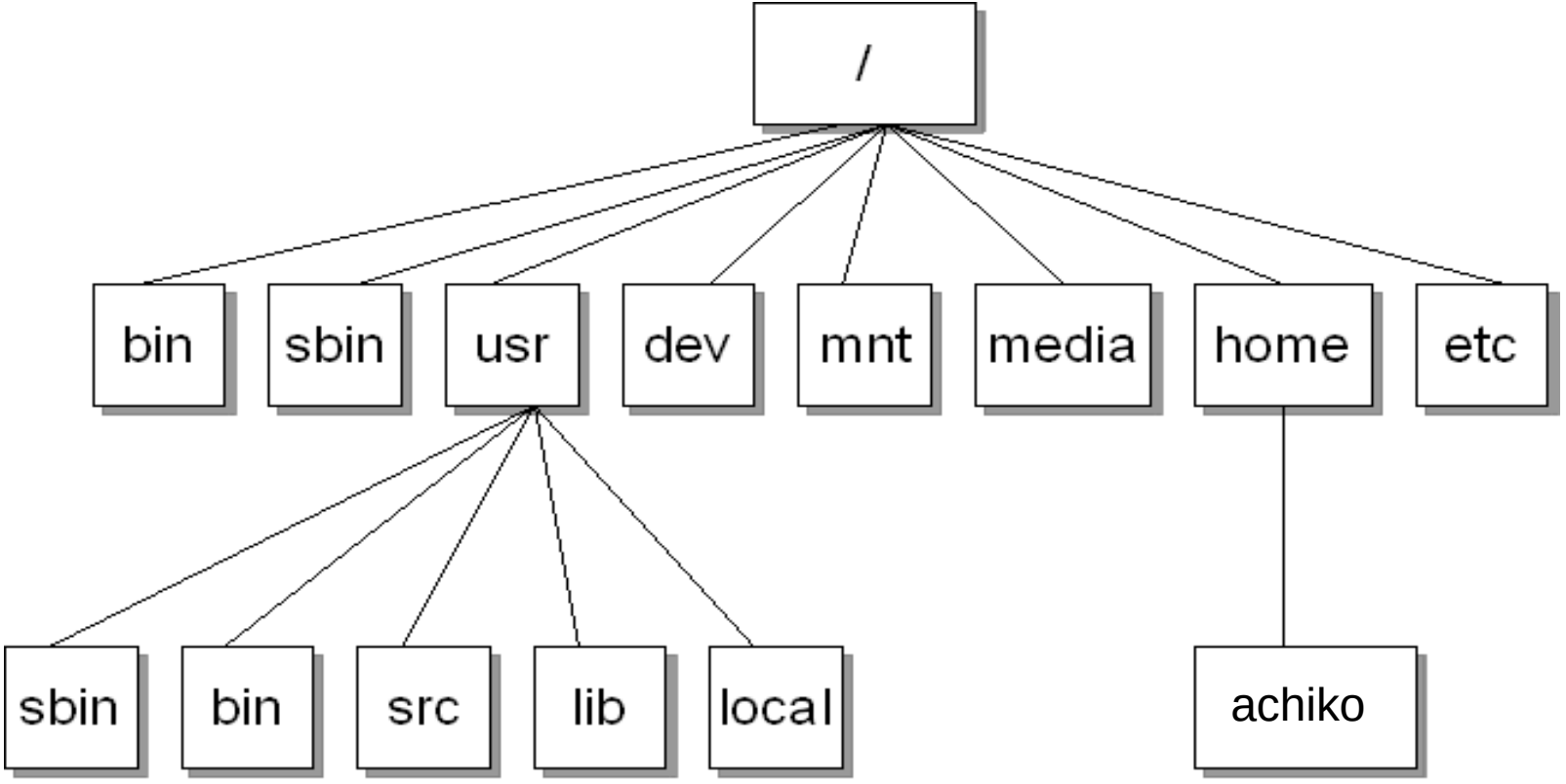
არჩილ ელიზბარაშვილი, 2024

ლინუქსის დისტრიბუტივები

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ლინუქსის სისტემის
ფაილების სტრუქტურა



LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

Filesystem Hierarchy Standard (version 3.0, released on 3 June 2015)

- გამშვები ფაილები:
  /bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin, /usr/local/sbin, /usr/X11R6/bin
- ბიბლიოთეკები: /lib, /usr/lib, /usr/local/lib
- ჩატვირთვისთვის საჭირო ფაილები: /boot
- მიერთებული მოწყობილობების ფაილები: /dev
- პირადი დირექტორიები: /home, /root
- /var, /etc, /tmp, /mnt, /media /opt, /proc, /run

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

დოკუმენტაცია

`man [-N] file` – ბრძანებათა ხაზის სახელმძღვანელო გვერდი. 9 სექცია : {1- მომხმარებლის ბრძანებები; 2- სისტემური გამოძახებები (ბირთვის ფუნქციები); 3- ბიბლიოთეკები; 4- სპეციალური ფაილები (როგორც წესი მდებარეობს `/dev`-ში); 5- კონფიგურაციის ფაილები; 6-თამაშები ; 7- სხვა; 8- სისტემის მართვის ბრძანებები; 9- ბირთვის ქვე-პროგრამები [სტანდარტს მიღმა]}; `/usr/share/man/`

`info file` – ეკრანზე გამოქვს განმარტება. `/usr/share/info/`

`README; FAQ; CHANELOG; HOWTO; /usr/share/doc/;`

`LDP (Linux Documentation Project); http://www.tldp.org`

`whatis name` – გამოაქვს name-ის მოკლე განმარტება სახელმძღვანელო გვერდიდან.
`whatis ⇔ man -f`

`apropos keyword` – გამოაქვს ყველა სახელმძღვანელო გვერდი, რომელიც განმარტებაც შეიცავს keyword-ს . `apropos ⇔ man -k`

`which cmd` – აბრუნებს ბრძანების სრულ გზას ; `whereis cmd` - აბრუნებს ბრძანებისა და მისი სახელმძღვანელო გვერდის სრულ გზას.

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024


```
ls; ls = ls .;  ls ..
ls file; ls directory;
ls (-l, -lu, -lc, -a,  -R, -d)
```

მაგენერირებელი სიმბოლოები : ფიფქი (*), კითხვის ნიშანი (?)

კვადრატული ფრჩხილები : [], [abc], [a-z], [^abc], [!abc]

ფიგურული ფრჩხილები : {}, {expr1, expr2, …}

```
ls {expr1, expr2, ...}* ; ls expr1*  ls expr2*  …
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ასო-ნიშნისთვის ფუნქციის დაკარგვა :

- უკან გადახრილი წილადის ხაზი (\)
- აპოსტროფი ('')
- ბრჭყალი ("")

```
ls --quoting-style=literal (Starting with GNU coreutils version 8.25 (released Jan. 2016), ls's default output quotes filenames with special characters), ls -N,--literal
touch file1 file2 ... ; {a..z}
tee file; tee -a file
cat file1 file2 ...
mkdir [-p] directory1 directory2;
cd [directory1]; .; ..
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

სტანდარტული შესასვლელი: `ls < file.input;` (ნომერი 0)

სტანდარტული გამოსასვლელი: `ls > file.output;` (ნომერი 1)

შეცდომები: `ls file 2>file.erreur;` (ნომერი 2)

`>>, <<, /dev/null, N>&M`

მაგალითები:

```
cat <file1 >file2; cat file1  1>file2  2>file3
cat << EOT > file; cat << text > file; cat >file (C^d)
cat file1>file2 2>&1;  >& file ⇔ &> file ⇔ >file 2>&1
tee file1 > file2
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ოპერატორები : ბრძანებათა გადაბმა, პირობითი გადაბმა :

```
command1 ; command2        command1 && command2
command1 & command2        command1 || command2
```

მილი: `command1 | command2; command1 | xargs command2;`
`more, less`

ბრძანების შეცვლა შედეგით : `` `command` ``, `$(command)`

გამოთვლა : `+, -, *, /, %`

`$[arithmetic_expression]; $((arithmetic_expression));`

`bc [-l]`

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024


```
cp file path
cp f1 f2; cp f1 f2 . . . dir/; cp -r dir1 dir2...dir/
mv file target
mv f1 f2; mv f1 f2 ... dir1 dir2 ... dir/
rm f1 f2 ...; rm -r dir1 dir2 ...;
rmdir dir1 dir2...
split -bn[bkm] file file_suffix
split -ln file file_suffix, -aN, -d, -CN
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024


```
cmp f1 f2; diff f1 f2; diff f1 r1, diff r1 f1, diff r1 r2;

patch file diff_file

find path expression

expression={-name nom, -type [fdlpbc], -size [-+]n[cbkMG],
        -empty, -uid(gid) n, -user/group nom, -perm xyz,
        -inum n, -links n, -atime +/-n, -amin +/-n, -anewer
        file, -ctime +/-n, -cmin +/-n, -cnewer file, -mtime
        +/-n, -mmin +/-n, -newer file, -fprint file}

find path expression -exec [ok] command {} \;[+]

-or, -and, \(, \)
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024


```
tail -[nc]N file (-f), head, cat, tac

cut -[fc] n-m,p ;-n,n-; file (-d), wc -[cwl]

sort -[fr] file (-n numerical, -k N : ალაგებს N სვეტის მიხედვით;
    ps aux | sort -n -k2 - ალაგებს PID-ით)

uniq -[ud] file, -c; sort file | uniq -u
tr a b, tr -c a b, tr -d X, tr -s X

tr soleil lune, s=>l, o=>u, l=>n, e=>e, i=>e, l=>e

tr A-Z a-z <==> tr '[:upper:]' '[:lower:]'

paste file1 file2 ... ; -s ; -d 'X'

join file1 file2
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024


```
date
date +%X, X={H, M, S, a, B, d, m, j, U, Y}, %t, %n}

date -d 'N days[moths, years] ago'; date -d N days, date -d
'2 jan 2008'

cal -[yj]

time cmd; /usr/bin/time -f "%E cmd"
```

(time არის შიდა ბრძანება (builtin); /usr/bin/time არ არის შიდა ბრძანება და ის თანხმდება ამ ფორმატს);

```
file filename

od (Octal Dump); -t {d,f,x,o,c...} = -d, -f, -x, -o, -c
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

PID, PPID ; `ps` ბრძანების სამი ტიპის ოფცია : UNIX ტიპის (ტირე), BSD ტიპის (ტირეს გარეშე), GNU-ს გძელი ფორმატი (ორი ტირე)

`ps {aux, ax, -e, -ef, -eF}` – ყველა პროცესი

`ps -u USER -U USER` – USER-ის პროცესები

`ps {-ejH, axjf}` – პროცესების ხე; `pstree (-p);`

`ps {-eo,-Ao,axo} pid,ni,psr,pcpu,stat,cmd` – მომხმარებლის მიერ განსაზღვრული ფორმატი (`-e=-A`)

`ps aux --sort pcpu` – დახარისხება cpu-ს დატვირთვის მიხედვით

`ps aux --sort pmem` – მეხსიერების დატვირთვის მიხედვით

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

პროცესის მდგომარეობა = {**R:** გაშვებული, მომუშავე პროცესი, **S:** დაძინებული პროცესი, რომელიც ინსტრუქციას ელოდება შესასრულებლად, **T:** შეჩერებული პროცესი, **Z:** ზომბი პროცესი. ის აღარაფერზე რეაგირებს (პროცესი დამთავრდა, თუმცა მშობელი პროცესის მიერ ვერ გაიწმინდა მეხსიერებიდან. როგორც წესი ასეთი შემთხვევები ცუდად დაწერილ პროგრამების გაშვებისას გხვდება)}

BSD-ს ფორმატის დამატებითი სიმბოლოები = {**<:** მაღალ პრიორიტეტულ პროცესს აღნიშნავს ანუ უდრო მეტი CPU დრო იხარჯება მის შესრულებაზე, **N:** დაბალ პრიორიტეტული, **L:** has pages locked into memory, **s:** სესიის ლიდერი, კონკრეტულ სესიაში წამყვანი პროცესი, **+:** წინა პლანზე არსებული პროცესი}

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

`kill -N PIDs, kill -l`

`Signals :{1 (HUP), 2(INT,C^c), 9(KILL), 15(TERM), 18(CONT), 19(STOP), 20 (TSTP,C^z)}`

SIGTERM არის პროცესის მოკვლის ყველაზე უსაფრთხო გზა (ნაგულისხმევი მნიშვნელობა). SIGHUP ნაკლებად უსაფრთხოა, ვიდრე SIGTERM. SIGKILL ; შენახვის გარეშე კლავს პროცესს.

`killall N name` – პროცესის მოკვლა სახელით

`xkill` – კლიენტის მოკვლა X resource-ით (მაუსით)

`pkill name` - SIGTERM სიგნალს გაუზგავნის ყველა პროცესს, რომლის სახელიც შეიცავს `name`-ს. (სიგნალის გაგზავნამდე, შეგიძლიათ შეამოწმოთ რომელ პროცესებს ჰქვიათ ასე: `pgrep -l name`)

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

`top [-d N]` (დაყოვნება N წამი), `-b -nN` (batch რეჟიმი N ცალი იტერაციით), `-i` (უგულებელყოფს უქმე (idle) პროცესებს), `-u user` (მომხმარებლის პროცესები), `-p pid1, pid2...` (მითითებული პროცესები)

Top-ის ბრძანებები = {**z:** Highlight, **c:** ბრძანების აბსოლუტური გზა, **k:** მოკვლა, **Shift+P:** დახარისხება cpu-თი, **r:** პრიორიტეტის შეცვლა, **h:** დახმარება, **q:** გამოსვლა}

პროცესის შესრულების პრიორიტეტი [-20,19]

`nice -n value (-value) command`

ნაგულისხმევი მნიშვნელობა = 0, -n -10 = --10

`renice value -p PID;`

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

`C-z, jobs [-l]:` გამოაქვს ფონურ რეჟიმში გაშვებული პროცესები `[PID]`-ც

`bg %N; fg %N; kill %N; cmd&; cmd& disown; nohup cmd&;`

cmd: უშვებს პროცესს ტერმინალის მიმდინარე bash-ში **წინა პლანზე** (პროცესი მოცემულია bash-ის წინა პლანზე გაშვეული ამოცანების სიაში და stdin, stdout და stderr კვლავაც მიბმულია ტერმინალზე);

cmd&: უშვებს პროცესს ტერმინალის მიმდინარე bash-ში **უკანა პლანზე** (პროცესი მოცემულია bash-ის უკანა გაშვეული ამოცანების სიაში და stdin, stdout და stderr კვლავაც მიბმულია ტერმინალზე);

cmd&disown: უშვებს პროცესს ტერმინალის მიმდინარე bash-ში **უკანა პლანზე**, მაგრამ პროცესი აღარ იქნება bash-ის jobs სიაში (პროცესი არ არის მოცემული bash-ის უკანა გაშვეული ამოცანების სიაში და stdin, stdout და stderr კვლავაც მიბმულია ტერმინალზე)

nohup cmd&: უშვებს პროცესს ტერმინალის მიმდინარე bash-ში **უკანა პლანზე**, მაგრამ პროცესი აღარ იქნება bash-ის jobs სიაში (პროცესი არ არის მოცემული bash-ის უკანა გაშვეული ამოცანების სიაში და stdin, stdout და stderr აღარ არის მიბმული ტერმინალზე)

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

`free -[bkm]` (byte,kbyte,mbyte), `-s N` (second), `/proc/meminfo`, `uptime`, `vmstat`

`Register` / `Buffer` / `Cache`:

კეში მეხსიერების უფრო მაღალ სიჩქარიანი წვდომის ნაწილია. განთავსებულია სტატიკურ მეხსიერებაში - SRAM (SRAM მოთავსებულია CPU-სა და RAM-ს შორის. თუმცა ძლავრი, თანამედროვე პროცესორები ხშირად L1 cache-ის მსგავსად პროცესორის შიგნითვე ათავსებენ ასეთ CPUL2 და L3 cache-ს), ბუფერი კი უფრო ნელი - დინამიური მეხსიერების (DRAM) ნაწილია. DRAM - ოპერტიული მეხსიერებაა. (მაგალითად DDR3-ც DRAM-ია).

ბუფერი ძირითადად გამოიყენება პროცესებს შორის მონაცემების შეტან/გამოტანისთვის. ამ დროს პროცესორი შედარებით ნაკლებად იტვირთება. კეში კი, სწრაფი წვდომის მიზნით, ხშირად გამოყენებადი მონაცემების ასლს აკეთებს.

კეში აგრეთვე შესაძლებელია იყოს მყარი დისკის ნაწილიც, ბუფერი კი მხოლოდ ოპერატიული მეხსიერების ნაწილია.

კლავიატურიდან წვდომა გვაქვს მხოლოდ ბუფერში არსებულ მონაცემებზე (მაგალითად ტექსტის რედაქტირებისას კოპირება და ჩასმა ხდება ბუფერში და ბუფერიდან), კეშზე არა.

კეშის გარდა პროცესორში არის უფრო სწრაფი წვდომის დროებით შესანახი ადგილი - რეგისტრი. რაც უფრო მეტია რეგისტრთა რაოდენობა, მით უფრო მძლავრია პროცესორი.

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ცვლადები, გარემო

ცვლადები: გარემოს ცვლადები ; ცვლადის შეცვლა შედეგით: `$variable, ${variable}`

```
variable=value; echo $variable; printenv; env; env variable=value cmd; set; unset; export variable; export -n variable;

USER, SHELL, PWD, OLDPWD, PATH=$PATH:/new_path, PATH=new_path:$PATH, LANG, HOME,

HISTFILE, HISTFILESIZE, PS1 (\h,\H,\u,\w,\W,\d,\t,\T,\@,\$,\n,\#,\!);
```

`echo -n, -e {\b,\010` – backspace, `\t,\011` – tab, `\n,\012` – new line, `\v,\013` – vertical tab, `\f,\014` – form feed (page break), `\r,\015` – carriage return, `\xHH` – HH=ascii hex, `\uHHHH` HHHH=unicode, `\0NNN` NNN=octale`}`

```
echo -e "\u10DA" => ლ

printf 'testing\012\011\011testing\014\010\012more testing\012\011\000\013\000even more testing\
012\011\011\011\012' | od -c

printf 'testing\012\011\011testing\014\010\012more testing\012\011\000\013\000even more testing\
012\011\011\011\012' >file;
```

`cat -A file` – გვაჩვენებს უხილავ სიმბოლოებსაც

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ბრძანება expr

a="abcABC123ABC123"; . = ერთი ასო-ნიშანი; .* = ყველა ასო-ნიშანი

ცვლადის სიგრძე : `expr length "$a"; expr $a : '.*'; expr|echo ${#a}`

ფრაგმენტის სიგრძე : `expr match "$chaine" "sous-chaine"; expr "$chaine" : "sous-chaine"`

`expr "$a" : '[a-z]*A' = 4 ; expr "$a" : '[a-z].*A' = 7`

ფრაგმენტის პოზიციის რიგითი ნომერი : `expr index "$chaine" "sous-chaine"`

`expr index "$a" '123' = 7`

ფრაგმენტის ამოღება : `expr substr "$a" position longueur ; expr ${chaine:position[:longeur]}`

`expr "$chaine" : "\(sous-chaine\)" ; expr match "$chaine" "\(sous-chaine\)"`

`expr "$chaine" : ".*\(sous-chaine\)" ; expr match "$chaine" ".*\(sous-chaine\)"`

დასაწყისიდან უმოკლესი/უშორესი : `expr|echo ${chaine#sous-chaine}; expr|echo ${a#a*C}; expr|echo ${a##a*C}`

ბოლოდან უმ./უშ. : `expr|echo ${chaine%sous-ch}; expr|echo ${a%b*c}; expr|echo ${a%%b*c}`

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ფრაგმენტის ჩანაცვლება სხვა გამოსახულებით :

```
expr|echo ${chaine/sous-chaine/remplacement}
expr|echo ${a/abc/AZERTY}; expr|echo ${a//abc/AZERTY};
expr|echo ${a/#abc/AZERTY}; expr|echo ${a/%abc/AZERTY};
```

დიდი და პატარა ასოების შეცვლა/გადანაცვლება :

```
expr|echo ${chaine^}, ${chaine^^} / ${chaine,},${chaine,,} / $
{chaine~}, ${chaine~~}
expr|echo ${a^}; expr|echo ${a^^}; expr|echo ${a,}; expr|echo ${a,,};
 expr|echo ${a~}; expr|echo ${a~~};
array=(This is some Text); echo "${array[@],}" ⇒ this is some text; echo "$
{array[@],,}" ⇒ this is some text;
echo "${array[@]^}" ⇒ This Is Some Text; echo "${array[@]^^}" ⇒ THIS IS
SOME TEXT; echo "${array[3]^^}" ⇒ TEXT
```

Ressource : http://www.tutorialspoint.com/unix/unix-vi-editor.htm

Vim=**V**i **Im**proved

vi fichier; vi -R fichier = view fichier; vi +15 fichier (კურსორი გადავა მე-15 ხაზზე თან)

ბრძანებათა რეჟიმი :

h(←), j(↓), k(↑), l(→)

L, H, gg(პირველი ხაზი), ngg, nG, G(ბოლო ხაზი), d, dd, ndd, [m]dn{h,j,k,l}

y, yy, nyy, [m]yn{h,j,k,l}, p, P, u, C^r, x, ~, r, R

:wq = :x = ZZ, :q!, :w, :r fichier, :!cmd, :[[x,y],%,$]s/expr1/expr2/[g]

:set nu[mber], :set nonu[mber]

/string, ?string, n, N

აკრეფის რეჟიმი : i, I, a, A, o, O

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ტექსტური რედაქტორი - emacs

Ressource : http://www.gnu.org/software/emacs/tour/

emacs file

C-f: →, C-p: ↑, C-n: ↓, C-b: ←, C-a: ^, C-e: $, C-v: შემდეგი გვერდი, M(Alt)-v: წინა გვერდი, C-x C-f: ახალი ფაილის გახსნა.

C-x C-c: გამოსვლა, C-x C-f: გახსნა, C-x C-s: შენახვა, C-x C-w: შენახვა როგორც, C-x C-b: სიის ნახვა, C-x b: ბუფერის გააქტიურება, C-x 1: სიის გაქრობა, C- ,C-@: მონიშვნა, C-g: გადანიშვნა, M-w: კოპირება, C-w: ამოჭრა, C-y: ჩასმა, C-_, C-/, C-x u: წინა ოპერაციის გაუქმება, M-x shell: shell-ის გამოძახება, M-x dired: ფაილების მართვა (file manager).

რეგისტრი: C-x r => s (რეგისტრში შენახვა), C-x r => i (რეგისტრიდან ჩასმა)

მოძებნა: C-s texte (C-s შემდეგზე გადასვლა)
ჩანაცვლება: M-% (y-შეცვლა, n-შემდეგზე გადასვლა, q-გამოსვლა, .-მიმდინარეს შეცვლა და გამოსვლა, !-ყველას შეცვლა)

ჩასმა: C-x i, ყველაფრის მონიშვნა: C-x h, ხაზზე გადასვლა: C-g, M-<, M->

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

რეგულარული გამოსახულება - grep

```
grep [options] expression fichier1 fichier2 …
```

ოფციები={-i(--ignore-case), -c(--count), -n(--line-number), -A NUM(--after-context=NUM), -B NUM(--before-context=NUM), -C NUM(--context=NUM), -E(--extended-regexp), -F(--fixed-strings), -r(--recursive), -v(--invert-match), -o(--only-matching), -l(--files-with-matches) PATH, -w(--word-regexp)...}

ძირითადი გამოსახულება={[abc],[a-z],[0-9],[^abc],[:alpha:],[:digit:],[:alnum:], [:lower:],[:upper:],[:blank:](space, tab),[:space:](tab, newline, vertical tab, form feed, carriage return and space),[:punct:],[:graph:](არა [:alnum:] ან [:punct:]),[:print:] ([:space:]და არა [:graph:]),[:xdigit:],r\?,r1\|r2,r\+,\{n\},\{n,\},\{n,m\}}
^, $, ., X***, \b, \B, \<**word**\>**

გაფართოებული გამოსახულება={r**?**, r1**|**r2, r**+**, **(**r**)**, r**{**n**}**, **{**n,**}**, **{**,n**}**, **{**n,m**}**}

```
grep -E = egrep ; grep -r = rgrep, grep -F = fgrep, zgrep, bzgrep
```

```
grep -e expression1 -e expression2 ... fichier1 fichier2 …
```

მაგ : `cat fichier1 | grep 'expr' - fichier2`: ეძებს როგორც სტ. გამოსასვლელში, ასევე ფაილებში

Ressource : http://www.grymoire.com/Unix/sed.html

sed (**S**tream **ED**itor)

- Sed კითხულობს ხაზს შესასვლელიდან (ფაილი, მილი ან სტ. შესასვლელი) და ინახავს მას გამოსახულების ბუფერში (pattern buffer).
- შემდეგ Sed ასრულებს ყველა მითითებულ ბრძანებას თანმიმდევრულად გამოსახულების ბუფერზე.
- ნაგულისხმევი მნიშვნელობით ეს ბრძანებები შესრულდება ყველა ხაზისთვის თუ კონკრეტული მისამართი არ არის მითითებული. შემდეგ Sed გამოიტანს მიღებულ შედეგს სტ. გამოსასვლელზე.
- ბოლოს გამოსახულების ბუფერში ყველაფერი იშლება.
- გამოსახულების ბუფერი არის მეხსიერების ნაწილი. მის გარდა, Sed მეხსიერებაში აქვს სარეზერვო ბუფერიც (hold buffer). თავიდან ის ყოველთვის ცარიელია.
- სარეზერვო ბუფერის მნიშვნელობა ციკლიდან ციკლამდე არ იშლება.
- პირდაპირ სარეზერვო ბუფერზე ბრძანებების შესრულება ვერ ხერხდება, თუმცა შესაძლებელია მონაცემების გაცვლა/კოპირება გამოსახულების ბუფერიდან სარეზერვო ბუფერში და პირიქით.

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

რეგულარული გამოსახულება - sed

Ressource : http://www.grymoire.com/Unix/sed.html

Commands : {d/D – შლის მონაცემებს გამოსახულების ბუფერში/პირველ შემხვედრ ახლა ხაზამდე, addra – ჩამატება ხაზის შემდეგ, addri – ჩამატება ხაზის წინ, p/P – ეკრანზე გამოტანას/ეკრანზე გამოტანას პირველ ნაწილს (ახალ ხაზამდე), q – გამოსვლა, s/…/…/[N,g,Ng, I(უგულებელყოფს დიდ და პატარა ასოებს)] – ჩანაცვლება, y/…/…/ – გარდაქმნა, :label – ჭდეს შექმნა, blabel – ჭდეზე გადასვლა, {…} - დაჯგუფება, = - დასამუშავებელი ხაზის ნომერი, [adr1,[adr2]]c text – adr-ის შეცვლა text-ით, x – გამოსახულებისა და სარეზერვო ბუფერების შიგთავსებს გადაანაცვლებს, h,H – გამოსახულების ბუფერის შიგთავსს აკოპირებს/ბოლოში მიამატებს სარეზერვო ბუფერში, g,G – აკოპირებს/ბოლოში მიამატებს სარეზერვო ბუფერის შიგთავსს გამოსახულების ბუფერში, n,N – წაიკითხავს/დაამატებს ახალ ხაზს გამოსახულების ბუფერში, w file, r file, e – შესრულება (shell-ის ბრძანება), \L/\U – გადააკეთებს პატარა/დიდ ასოებად}

Options : {-n, --quiet, --silent: ნაგულისხმევი მნიშვნელობით ბეჭდავს გამოსახულების ბუფერს, -i[SUFFIX], -r: გაფართოებული რეგულარული გამოსახულება}

ADDRESS={N,/expression/} ; ADDR1[,ADDR2]; ADDR1,+N ; N~M ;

```
sed -e instruction1 -e instruction2 … f1 f2 ...;  sed -f programm file
```

`sed '2,+5d' fichier` – გამოიტანს მე-2დან 5 ხაზის გარდა დანარჩენს ;

`sed '3q' fichier` – გამოიტანს პირველ 3 ხაზს.

`sed -n '/un/,+3p' fichier` – გამოიტანს 3 ხაზს იმ ხაზიდან დაწყებული, რომელიც შეიცავს "un"-ს

`sed 's/.*/\U&/' fichier` – დიდ ასოებად გადააკეთებს;

`sed -n '/un/,/deux/ !p' fichier` – გამოიტანს იმ ხაზების გარდა სხვებს, რომლებიც "un"-ის შემცველი ხაზიდან დაწყებული "deux"-ს შემცველ ხაზამდეა.

`sed '1itexte' fichier` – პირველი ხაზის წინ ჩასვამს texte-ს

`sed '$atexte' fichier` – ბოლო ხაზის შემდეგ ჩასვამს texte-ს.

`sed '2,7s/un/deux/3g' fichier` – მე-2დან მე-7 ჩათვლით ხაზებში შეცვალოს პირველი 3 შემხვედრი "un" "deux"-თი.

`sed 's/un/(&)/Ig' fichier` – შეცვალოს "un" "(un)"-ით ყველგან (ასოების სიდიდე უგულებელყოფილია)

`sed -n '$=' fichier` – ხაზების რეოდენობა.

`sed '/un/,5c texte' fichier` – შეცვალოს "un"-ის შემცველი ხაზიდან მე-5 ხაზის ჩათვლით texte-ით.

`sed -e 'y/abc/xyz/' -e 'w new_f' f` – ჩაანაცვლოს a x-ით, b y-ით, c z-ით და შეინახოს ფაილში new_f.

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

sed
მაგალითები

`sed '$r new_f' f` – ბოლოში ჩასვას `new_f`-ის შიგთავსი

`sed 's/\(.*\),\(.*\),\(.*\)/\2:\1/'` – შეცვალოს მთლიანი ხაზი \2:\1 ცვლადებით.

`sed -i.bak '1e date' f` – ჩასვას `date` ბრძანების შედეგი 1 ხაზის წინ და შეინახოს `f.bak`-ში.

`sed ':a;N;$!ba;s/\n/ /g' f` – წაშალოს ახალ ხაზზე გადასვლა (1.(**:a**) შექმნას ჭდე **a**. 2.(**N**) დაამატოს შემდეგი ხაზი გამოსახულების ბუფერს. 3.(**$!ba**) ციკლი 'a'-სთან დაბრუნებით ფაილის ბოლომდე ($! ნიშნავს არა ბოლო ხაზზე) 4.(**s/\n/ /g**) შეცვალოს ყველა ახალ ხაზზე გადასვლა გამოტოვებით) = `sed -e ':a' -e 'N' -e '$!ba' -e 's/\n/ /g'`.

`sed '/Windows/{N;d} f` – წაშალოს `Windows` შემცველი ხაზები და მისი შემდეგი ხაზიც

`sed 'G' f` – ერთი ხაზის გამოტოვებით (G;G – ორი ხაზის გამოტოვებით)

`sed -n '1!G;h;$p'` – გამოიტანს ეკრანზე შიგთავსს პირუკუ (1.პირველი ხაზი განთავსდება სარეზერვო ბუფერში. 2.მე-2 ხაზიდან სარეზერვო ბუფერის შიგთავსი დაემატება გამოსახულების ბუფერს. (ყურადღება: მე-2 ხაზი არის გამოსახულების ბუფერში, ხოლო პირველი არის სარეზერვო ბუფერში). 3.ასე პირველი და მეორე ხაზები შეტრიალდა და ეს უკვე გადავა სარეზერვო ბუფერში. 4.ამ პროცედურის გამეორება ბოლო ხაზამდე. 5.ბოლო ხაზზე კი დაამატებს სარეზერვო ბუფერის შიგთავსს გამოსახულების ბუფერსს და დაბეჭდავს)

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

რეგულარული გამოსახულება - awk

Ressource : http://www.tutorialspoint.com/awk/index.htm

AWK: Alfred **A**ho, Peter **W**einberger, and Brian **K**ernighan ; AWK – AT&T ლაბორატორიაში შექმნილი ორიგინალი AWK. NAWK – AWK-ს ახალი და გაუმჯობესებული ვერსია (AT&T). GAWK – AWK-ს GNU-ს ვერსია. ყველა GNU/Linux დისტრიბუტივში შედის GAWK. ის სრულიად თავსებადია AWK-სა და NAWK-სთან.

```
awk  'BEGIN {print "debut"}      {print "toute la ligne = $0"}END {print "fin"}'  Filename
```

ცვლადები : {$0 – მთლიანი ხაზი, $N - მე-N ველი, FS - ველებს შორის გამყოფი, NF – ველების რაოდენობა, NR – მიმდინარე ხახის ნომერი, FNR – იგივეა, რაც NR, ოღონდ ეხება მიმდინარე ფაილს, OFS – გამყოფი გამოტანისას, RS – ხაზებს შორის გამყოფი შეტანისას, ORS - ხაზებს შორის გამყოფი გამოტანისას, ARGC - არგუმენტების რაოდენობა, ARGV – მასივი, რომელიც ინახავს არგუმენტებს, FILENAME – მიმდინარე ფაილის სახელი, ENVIRON – გარემოს ცვლადების მასივი}

ოპერატორები : {+,-,*,/,%,^,=,+=,-=,*=,/=,%=,^= - არითმეტიკული ოპერაციები, მინიჭება, ++i, i++, --i, i-- - პრე/პოსტ ინკრიმენტი/დეკრიმენტი, ==,!=,<,>,<=,>= - შედარება, expr1||expr2, expr1&&expr2, ! expr – ლოგიკური ოპერაციები, condition expr ? Statement1 : statement2 – სამმაგი პირობა/ternary (როდესაც პირობა ჭეშმარიტია, მაშინ შესრულდება statement1, ჩინააღმდეგ შემთხვევაში statement2), s1="un,";s2="deux" ; s3=s1 s2 ; print s3 – სტრინგის კონკატენაცია, N [!]~ /expression/ - რეგულარული გამოსახულება}

მასივები : {array_name[index] = value – მასივის შექმნა; delete array_name[index] – მასივის ელემენტის წაშლა, array["0,0"] = 100 – მრავალგანზომილებიანი მასივი}

პირობები : { if (condition) action - If ; if (condition) action-1; else action-2 – If-Else, if (condition 1) action-1 ; else if (condition 2) action-2 ;... - If-Else-If}

ციკლები : {for (initialisation; condition; increment/decrement) action, while (condition) action, do action while (condition)} ; awk -f programme_file fichier

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

AWK
მაგალითები

`awk '{print;}' fichier` - Awk ნაგულისხმევი მნიშვნელობით ბეჭდავს ფაილის ყოველ ხაზს.

`awk -F: '{print $3 \t $1}' fichier` – გამოიტანს მე-3 და პირველ სვეტს.

`awk 'END {print NR}' fichier` – გამოიტანს ხაზების რაოდენობას.

`awk '/bonjour/ {print $0}' fichier` – გამოიტანს ხაზებს, რომლებიც შეიცავს გმოსახულებას `bonjour`.

`awk '$3>100 && $1~/bonjour/ {print $0}' fichier` - შეიცავს `bonjour`-ს და მე-3 სვეტის მნიშვნელობა>100

`awk '$1 !~ /^bonjour/ {print $0}' fichier` – ხაზი, რომლის პირველი ველი არ შეიცავს `bonjour`-ს.

`awk -v var1=X -v var2=Y 'BEGIN {printf "Variables = %s, %s\n", var1, var2}'` – გადაცემული ცვლადები.

`awk '/bonjour/{++cnt} END {print "Count = ", cnt}' fichier` – ითვლის მოცემულ გამოსახულების შემცველ ხაზებს.

`awk 'length($0)' > 15 fichier` – ბეჭდავს ხაზებს, რომლის სიგრძეც 15-ზე მეტია.

`awk 'BEGIN {for (i=0;i<ARGC;++i){printf "ARGV[%d]=%s\n",i,ARGV[i]}}' a b c` – ბრძანები და არგუმენტები.

`awk 'BEGIN { m=5; for(i=1;i<=m;i++) { n=5; for(j=1;j<=n;j++) { array[i,j]=j%i; printf "%d ",array[i, j]} printf "\n" } }'` - ბეჭდავს მრავალგანზომილებიან მასივს.

`awk 'BEGIN { num = 22; if (num % 2 == 0) printf "%d is even number.\n", num; else printf "%d is odd number.\n", num }'` - კენტია თუ ლუწი.

`awk 'BEGIN { pi=3.1415; i=-pi; while (i<=pi) { i=i+0.001; printf "sin(%f)=%f\n", i, sin(i) > "file.txt"}}'` - გამოითვლის `sin(x)`-ს, სადაც `x=[-π,π]` და შედეგს გადაიტანს ფაილში `file.txt`

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024


```
at; atd; at [time] {HH:MM, [MMDDYY, MM/DD/YY, DD.MM.YY],
now[+] [minutes, hours, days, weeks], teatime, midnight},
/usr/share/doc/at/timespec (ex : at 3:20pm tomorrow), at -f
cmd_list time; atq; atrm (at -d); /var/spool/…
```

თუ `/etc/at.allow` არსებობს, მაშინ მასში ჩაწერილ მომხმარებლებს შეუძლიათ გამოიყენონ `at`. თუ არა, მაშინ `/etc/at.deny` მოიძებნება და იქ ჩაწერილ მომხმარებლების გარდა სხვებს შეეძლებათ გამოიყენონ `at`. თუ არცერთი ფაილი არსებობს, მაშინ მხოლოდ root-ს შეუძლია გამოიყენოს `at`.

`batch` – შეასრულებს ამოცანას, როდესაც სისტემის დატვირთვა მისცემს ამის საშუალებას.

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ამოცანების გაშვების ავტომატიზირება

Cron; /etc/cron.allow => /etc/cron.deny;
თუ არცერთი ფაილი არსებობს, მაშინ ყველა მომხმარებელს შეუძლიათ გამოიყენონ crontab ან მხოლოდ root-ს (Debian-ში ყველა მომხმარებელს შეუძლიათ);

```
*   ,   n-m   n-m/p
```

```
crontab (-e,-l,-u user), /var/spool/cron/, EDITOR/VISUAL,
/etc/crontab
```

წუთი(*0-59*) საათი(*0-23*) თვის დღე(*1-31*) თვე(*1-12*) კვირის დღე(*0-7*) ბრძანება. (0=7=კვირა)

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

`tar [-cxftvrzj] fichier.tar[.gz/bz2] f1 f2 ... r1 r2 … ;`

`tar cvjf arch.tar.bz2 f… r…` - ქმნის არქივის ფაილს tar.bz2

`tar xvf arch.tar -C dest_rep/` - ამოაარქივებს მოცემულ დირექტორიაში

`tar -xvf arch.tar f1/"f1" "f2"` - ამოაარქივებს მხოლოდ მითითებულ ფაილებს

`tar -xvf arch.tar –wildcards '*.sh'` - ამოაარქივებს მხოლოდ მითითებულ ფაილებს

`tar -cjf - arch.tar | wc -c` – გამოიტანს tar, tar.gz და tar.bz2 ფაილების ზომას (KB-ში)

`find … | tar -cvf arch.tar -T -` - მოძებნის ფაილებს და დააარქივებს

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

gzip (Lempel-Ziv coding LZ77), gunzip=gzip -d (.gz)

bzip2(Burrows-Wheeler block sorting text compression algorithm and Huffman coding), bunzip2, bunzip2=bzip2 -d (.bz2),
[z/bz]cat,[z/bz]diff,[z/bz]grep,[z/bz]less,[z/bz]more

gzip/bzip2 -c file1.txt > files.[gz/bz2] – შეკუმშავს ფაილს და შეინახავს ორიგინალს

gzip/bzip2 -c file2.txt >> files.[gz/bz2] – მიაჯრის ფაილს

gzip -r * - შეკუმშავს ყველა ფაილს რეკურსიულად (დირექტორიებს არა)

gzip -tv f.txt.gz – მთლიანობის ტესტი

time gzip/bzip2 -1/-9 f.txt – შეკუმშვის დონის განსაზღვრა (უსწრაფესი/უკეთესი)

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

Ressource: http://www.slashroot.in/how-are-passwords-stored-linux-understanding-hashing-shadow-utils

მომხმარებლები (/etc/passwd, /etc/shadow), ჯგუფები (/etc/group, /etc/gshadow); {Md5, Blowfish, SHA{256,512}}; (password+salt), pwconv; pwunconv; grpconv; grpunconv; passwd user; gpasswd group

უფლებები: rwx, suid, sguid, t; chmod [ugoa]+/-[rwxst] file, [0-7]; umask

su [user]; su - ⇔ su -l ⇔ su --login; whoami; chown user[./: group] file; chgrp group file

who; w; loginctl; loginctl show-user username; /etc/issue[.net], /etc/motd

sudo command ; /etc/sudoers ; visudo; sudo -s

User/%group ALL=(ALL) ALL – (User/%group: მომხმარებლის/ჯგუფის სახელი, ვისაც sudo-ს გამოყენების უფლება აქვს; ALL: sudo წვდომის დაშვება ნებისმიერი ტერმინალიდან (კომპიუტერიდან); (ALL): sudo-თი ბრძანების გაშვება, როგორც ნებისმიერი მომხმარებელი; ALL: ყველა ბრძანების გაშვების უფლება)

User/%group ALL=PATH_to_cmd1,PATH_to_cmd2 – მომხმარებლს/ჯგუფს შეუძლია მხოლოდ ამ ბრძანებების გაშვება.

User/%group ALL= NOPASSWD : PATH_to_cmd – მომხმარებლს/ჯგუფს შეუძლია ბრძანების გაშვება პაროლის გარეშე.

Ressource: http://www.slashroot.in/how-are-passwords-stored-linux-understanding-hashing-shadow-utils

```
useradd {-c comment,-g group,-u UID,-s shell,-d dir,-k /etc/skel,-e date, -m} user
```

```
useradd -p $(mkpasswd -m md5 password_text) user; useradd -p $(echo pass | openssl passwd -1 -stdin) user
```

`userdel [-r] user ; usermod (-l, -c, -g, -s, -d, -u) user ; useradd -G group1 group2 … user` (მომხმარებლის დამატება რამდენიმე ჯგუფში) ; `useradd/usermod -e /2019-01-31 user` (მომხმარებილის ანგარიშის ვადის შექმნა/დაყენება) ;

`adduser user` - adduser არის perl-ის სკრიფტი, რომელიც იყენებს **useradd** გამშვებს

`newusers users_file.txt` – რამდენიმე მომხმარებლის შექმნა

`/etc/default/useradd (/etc/adduser.conf), /etc/login.defs; useradd -D`

`newgrp group1` (group1-თან მიერთება); `id ; groups`

`gpasswd (-a=მომხმარებლის დამატება, -d=მომხმარებლის ამოღება`
`ex: gpasswd -a user group = usermod -a -G group user)`

`groupadd (-g GID) group, groupdel group, id user ; groupmod (-n nom, -g GID) group`

`adduser user sudo / usermod -aG wheel user` – დაამატო მომხმარებელი sudoer-ებში debian/centos.

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

დაბლოკვა: `passwd -l user`, `usermod -L user`; განბლოკვა: `passwd -u user`, `usermod -U user`
პაროლის წაშლა: `passwd -d user`; დაშიფრული პაროლის გამოსახულება - {! ან null – არ აქვს პაროლი ; * - ანგარიში არ არის გააქტიურებული (disabled)', !<encrypted password> - ანგარიში დაბლოკილია ; !! - პაროლი არასოდეს დადებულა}

`chage [-l] [-m min_days] [-M max_days] [-W warn] [-I inactive] [-E expire] user`

min_days – დღეების ის მინიმალური რაოდენობა, რომელიც უნდა გავიდეს, რათა მომხმარებელმა შეძლოს საკუთარი პაროლის შეცვლა წინა შეცვლის შემდეგ. 0 ნიშნავს, რომ როდესაც სურს მაშინ შეცვლის პაროლს მომხმარებელი.

max_days - დღეების ის მაქსიმალური რაოდენობა, რომლის დროსაც პაროლი ვალიდურია. შემდეგ მომხმარებელმა უნდა შეცვალოს პაროლი.

Warn – დღეების ის რაოდენობა, როდესაც მომხმარებელი გაფრთხილდება, სანამ პაროლს ვადა გაუვა.

Inactive – პაროლის ვადის გასვლის შემდეგ დღეების რაოდენობა, რომლის განმავლობაშიც პაროლი მაინც მიიღება (შემდეგ შესვლაზე უნდა შეიცვალოს პაროლი). თუ არა და მომხმარებელი დაიბლოკება.

Expire – მომხმარებლის ანგარიშის გასვლის თარიღი. ის შეიძლება გამოხატული იყოს 1970 წლის 1 იანვრიდან გასული დღეებით ან ასე - YYYY-MM-DD ; -1 ნიშნავს, რომ ვადა არასოდეს გაუვიდეს ანგარიშს.

`chage -d 0 user` – აიძულებს მომხმარბელს შემდეგ შესვლაზე შეცვალოს პაროლი;

`chage -I 5 user` - აიძულებს მომხმარბელი დაიბლოკოს 5 დღიანი პასიურობის შემდეგ; `-1` - არასოდეს;

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

- საინსტალაციო პაკეტი:
  - კონფიგურაციის ფაილები
  - საჭირო ბიბლიოთეკების სია
  - ფუნქციონირებისათვის საჭირო პროგრამების სია
  - პაკეტში შემავალი ფაილების სია
  - დაყენება/ამოშლის სკრიფტები
- `paquet.tar.gz[bz2] tgz ...`
  - `cd source`
  - `./configure => Makefile`
  - `make`
  - `make install`
  - INSTALL, README ...

<name>-<version>-<release>.<architecture>.rpm

დაყენება: rpm -i package.rpm (-i,--install; -v: მეტი ინფო. (verbose); -h: ბეჭდავს #-ებს (hash marks))

განახლება: rpm -Uvh package.rpm (-U,--upgrade: დააყენებს თუ არ არის დაყენებული; -F, --freshen: თუ არა და განაახლებს)

ამოშლა: rpm -evv package (-e, --erase)

შემოწმება: rpm -Vv[p] package[.rpm] (დაყენებული პაკეტის ფაილების ინფორმაციას ადარებს rpm ბაზის მონაცემებს)

შეკითხვა: rpm -q package (-q, --query: ინფორმაციის გამოთხოვა, rpm დაყენებულია თუ არა)

ყველა დაყენებული პაკეტის გამოტანა: rpm -qa

შემოწმება: rpm -q[p]X package[.rpm] X={R- დამოკიდებულება [დაყენებამდე], c- კონფიგურაციების ფაილები, i- ინფორმაცია, l- ყველა ფაილი, d- დოკუმენტაცია, s- მდგომარეობა (normal, non install, remplac), f,--whatprovides file- პაკეტი)}

rpm-ის ვერსია: rpm --version

/etc/yum.conf, /etc/yum.repos.d/

```
yum [-y] OP [package]

   OP={install, remove/erase, info, update, list [installed], search, check-
update, provides/whatprovides file_path(or */file)}
```

yum grouplist (გამოიტანს პაკეტების ყველა შესაძლო დაჯგუფებას)

yum groupinstall/groupupdate/groupremove 'group_name'

yum repolist [all] (გამოიტანს გააქტიოურებულ [ყველა] Yum რეპოზიტორებს)

yum --enablerepo=epel install package (დააყენებს მითითებული რეპოზიტორიდან)

yum clean all (გაწმენდს Yum-ის კეშს /var/cache/yum/-ში)

yum history (ანახებს Yum-ის ისტორიას)

yum shell (yum shell => info setup – ინფორმაცია დაყენებული პაკეტების შესახებ « setup »)

yumdownloader [--resolve] package

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

პაკეტების ინსტალაცია
Dpkg (Debian Packages)

Stable, testing, unstable (Still In Developpement) (buster, stretch, jessie, wheezy, squeeze, leny, etch, woody...)

Main, non-free, contrib; `<name>_<version>-<release>_<architecture>.deb`

`dpkg, /var/lib/dpkg; /var/lib/dpkg/available` – შეიცავს ყველა დაყენებულ პაკეტებს

`-i prog.deb:` დაყენება

`-r/-P monsoft.deb :` ამოშლა/თავის ფაილებიანად.

`-l :` გამოიტანს ყველა დაყენებულ პაკეტს; `-s paquetage` – პაკეტის სტატუსის რეპორტი

`-S pattern:` გამოიტანს პაკეტს, რომელიც შეიცავს გამოსახულებას = `dlocate pattern`

`-L paquetage:` გამოიტანს პაკეტის ფაილებს (თუ დაყენებულია) და მათ ადგილმდებარეობას

`-c paquetage.deb:` გამოიტანს პაკეტის ფაილებს

`-p (--print-avail) paquetaqe:` გამოიტანს პაკეტის განმარტებას
`dpkg --configure` – დააკონფიგურირებს პაკეტს (მის კონფიგურაციის ფაილს)

`dpkg-reconfigure <package_name>` - ხელახლა დააკონფიგურირებს (მაგ: `dpkg-reconfigure locales`)

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

პაკეტების ინსტალაცია
APT (Advanced Package Tool)

APT (Advanced Package Tool) ; /etc/apt/sources.list

aptitude, synaptics, tasksel

apt-get [-y] OP [package] ; apt

OP={install, remove, purge, update, upgrade, dist-upgrade, clean (/var/cache/apt/archives), autoclean, downlaod (გადმოწერს deb პაკეტს), source (გააქტიურებული უნდა იყოს deb-src), autoremove ...}

apt-get install prog1 prog2- – დააყენებს 'prog1'-ს და ამოშლის 'prog2'-ს

apt-get remove prog1 prog2+ – ამოშლის 'prog1'-ს და დააყენებს 'prog2'-ს

apt-get --reinstall prog – ხელახლა დააყენებს 'prog'-ს

apt-get -s install '*prog*' – დაყენების სიმულაცია

apt-get -f install – დაზიანებული პაკეტების გამოსწორება

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

პაკეტების ინსტალაცია
APT (Advanced Package Tool)

`apt-cache search pattern` – ეძებს დასაყენებელი პაკეტების სიაში გამოსახულებას

`apt-cache show paquet` – გამოაქვს პაკეტის განმარტება

`apt-cache depends paquet` – გვანახებს პაკეტის დამოკიდებულებას

`apt-cache pkgnames` – გამოაქვს APT-სთვის ნაცნობი პაკეტების სახელები

`apt-file update ; netselect-apt`

`apt-file search <fichier>` : გამოაქვს პაკეტის სახელი, რომელშიც შედის <fichier>

`apt-file list <package>` : გამოაქვს პაკეტის შიგთავსი

`alien -d [--to-deb]` : გადააკეთებს პაკეტს deb-ად

`alien -r [--to-rpm]` : გადააკეთებს პაკეტს rpm-ად

`alien -t [--to-tgz]` : გადააკეთებს პაკეტს tgz-ად

www.kernel.org

მონოლითური; მოდულური => /lib/modules/<version-du-noyau>;

lsmod; rmmod [-s] (შეცდომები გადავა syslog-ში და არა შეც. გამოსასვლელზე) module; insmod [-s] module; modinfo module

modprobe -r (ამოშლა) module ; -l (სია) ; -s(syslog-ში) ; -a (ყველა მოდული)

depmod – ქმნის მოდულებს შორის დამოკიდებულებების მონაცემთა ბაზას- /lib/modules/<version-du-noyau>/modules.dep

modprobe – ჩატვირთავს არგუმენტად გადაცემულ მოდულებს და მათ დამოკიდებულებებსაც modules.dep-დან

/etc/modules – მოდულების ჩატვირთვა სისტემის ჩართვისას

cd /usr/src/linux_new

make config/menuconfig/xconfig => .config; make oldconfig
make modules – დინამიური მოდულების კომპილაცია
make modules_install – შესაბამის ადგილას გადატანა /lib/modules/<new_uname>

make install => /boot/{vmlinuz-x.y.z,System.map-x.y.z,initrd.img-x.y.z,config-x,y,z}

make menuconfig && time make deb-pkg && dpkg -i ../*.deb && shutdown -r now; uname -r

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

მყარი (ხისტი) დისკის სტრუქტურა

CHS – Cylinder-Head-Sector. ცილინდრი-ბილიკი-სექტორი

CHS მყარი დისკის დამისამართების ძველი მეთოდი.

დღეს, ის მაინც გამოიყენება სხვადასხვა პროგრამებში.

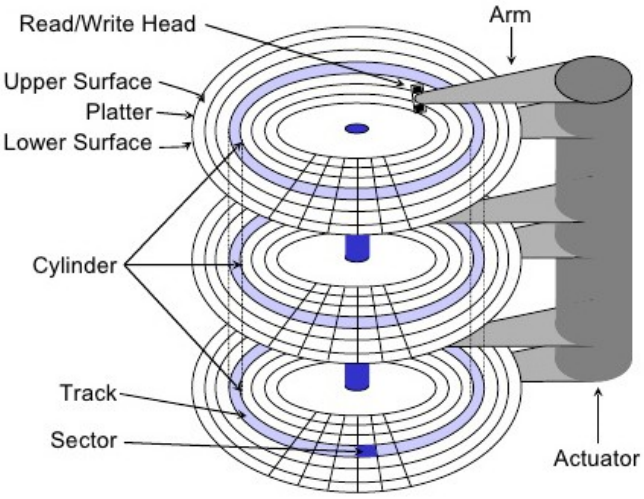
ტერმინოლოგია:

Tracks: the concentric rings

Each track is divided into multiple sectors

Cylinder: a hard disk consists of one or more platters with a read-write head is located on each side

of the platter. A vertical section on the corresponding ring across all platters and sides is called a cylinder.



სექტორები შეიძლება დამისამართდეს CHS კოორდინატებში 8 გიგაბაიტამდე :

C: ცილინდრი (0-1023). **H**: ბილიკი (0-254). **S**: სექტორი, (1-63)

რა თქმა უნდა, მითითებული ინტერვალი არ ასახავს ფიზიკურ ფაქტს. 0-255 ბილიკიანი ხისტი დისკი არ არსებობს. ეს მაქსიმალური მნიშვნელობა ერთხელ იქნა გამოყენებული BIOS-ის მიერ ხისტი დისკის დამისამართებისთვის. დისკის კონტროლერი შემდეგ აკონვერტირებს არსებულ მახასიათებლებს. CHS დამისამართება ძირითადად გამოიყენება დანაყოფების მართვის პროგრამებში. **C** და **H** მნიშვნელობა იწყება 0-ით, **S** კი 1-ით. ანუ, პირველი სექტორი **CHS** დამისამართებაში არის 0/0/1. თითოეული სექტორი 512 ბაიტის ზომისაა (4 096 ბაიტიანი სექტორი 2010 წლიდან გახდა ხელმისაწვდომი). ამ მიდგომით ხისტი დისკის მაქსიმალური ზომა დაახლოებით იქნება 8 GB (1024*255*63*512 ≈ 8 GB).

LBA – Logical Block Addressing, ლოგიკური ბლოკების დამისამართება

7,844 გ-ზე მეტი ზომის ხისტი დისკებისთვის CHS დამისამართების მხარდაჭერა საჭიროა გახდა მაინც თავსებადობისათვის (მინიმუმ ჩარტვირთვის დროს). თანამედროვე BIOS-ებს შეუძლიათ Int13h წყვეტით LBA რეჟიმზე გადაეწყოს სისტემა. LBA-ს გამოყენებით კი დამისამართება წრფივია, ბლოკების ათვლა იწყება 0-დან. 1 LBA ბლოკი შეესაბამება 1 CHS სექტორს.

LINUX სისტემები

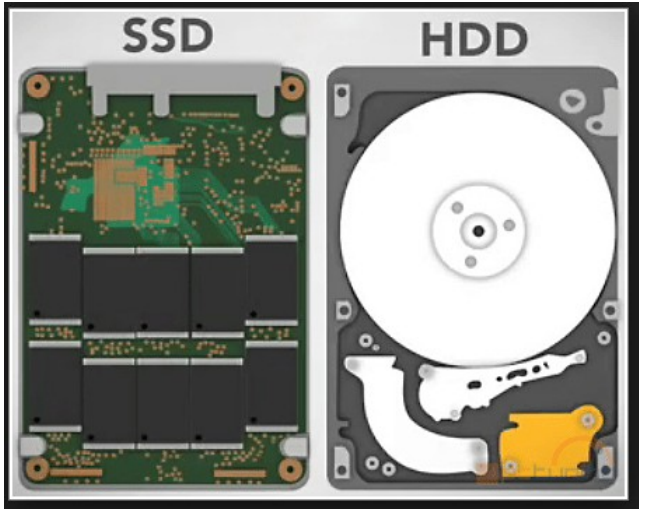
არჩილ ელიზბარაშვილი, 2024

SSD დისკები

SSD (Solide State Drive) არის non-volatile (არა არამდგრადი) დისკი. ანუ, დენის კვების არ მიწოდების მიუხედავად მონაცემი ინახება განსხვავებით RAM-ისგან (volatile).
HDD-ში მონაცემები ინახება მაგნიტურ დისკებზე, SSD-ში flash-memory chip-ებზე. სწორედ flash chip არის non-volatile. flash chip ასე გამოიყურება:

SSD დისკები გაცილებით ძვირია, ვიდრე HDD დისკები (ასევე, ძვირია USB flash მეხსიერებებთან შედარებით).

ფიზიკურად ასე გამოიყურება SSD და HDD :



HDD ცალკეული მექანიკური ნაწილებისგან შედგება, SSD-ს flash chip-ები კი ერთიანი ნაწილია. შესაბამისად, ძლიერ ფიზიკურ ზემოქმედებების მიმართ (მაგალითად, დავარდნა ან დარტყმა) SSD მდგრადია, HDD – არა.

SSD დისკებზე მონაცემების ჩაწერა/წაკითვის სიჩქარე გაცილების სწრაფია ვიდრე HDD დისკებზე. თუმცა SSD დისკები გაცილებით ძვირია.

HDD დისკების მუშაობისას პატარა ხმაური ისმინება. SSD-ზე საერთად არა. ელექტრო ენერგიის მოხმარების მხრივ - SSD გაცილებით ნაკლებს მოიხმარს, ვიდრე HDD.

SSD-ზე მონაცემის ჩაწერების რაოდენობა შეზღუდულია. HDD-ზე - არა.

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

MBR (Master Boot Record) პირველად შემოთავაზებული იყო 1983 წელს IBM PC DOS 2.0-ში. GPT (GUID Partition Table) არის დანაყოფების ცხრილის განლაგების ახალი სტანდარტი HDD-სა (Hard Disk Drive) და SSD-სთვის (Solid State Drive). დანაყოფების ამოსაცნობად ის იყენებს GUID-ს (Globally Unique IDentifier), რომელიც წარმოადგენს 32 სიმბოლოიან Unicode დასახელებას. ამავროულად ის UEFI-ის (Unified Extensible Firmware Interface) სტანდარტის ნაწილია. 2010 წლიდან უმეტეს ოპერაციულ სისტემებს აქვთ GPT-ის მხარდაჭერა, ზოგიერთს კი მხოლოდ ეს აქვს. GPT-ს ლოგიკური ბლოკების დამისამართების 64 ბიტიანი სისტემა აქვს, რაც იმის საშუალებას იძლება რომ აღიქვას 2^{64} რაოდენობა სექტორიანი დისკები. 512-ბაიტიანი სექტორის შემთხვევაში ზომის მაქსიმუმი არის 9.4 ZB (9.4×10^{21} bytes) ან 8 ZiB. GPT-ს აქვს :

- 2 TiB-ზე მეტი ზომის დისკების მხარდაჭერა
- 2 TiB-ზე მეტი ზომის დანაყოფების მხარდაჭერა
- 4 -ზე მეტი რაოდენობა დანაყოფების მხარდაჭერა (მნიშვნელობა არ აქვს პირველადი, გაფართოებული თუ ლოგიკური)
- იყენებს GUID-ს, რაც ორი ერთნაირი დასახელების არსებობის ნაკლებ შანსს იძლევა სხვადასხვა დანაყოფებისთვის.
- იყენებს მხოლოდ LBA დამისამართებას MBR-სგან განსხვავებით, რომელიც ორივეს, LBA და CHS, იყენებს.
- დანაყოფების ცხრილს ინახავს როგორც დისკის დასაწყისში ასევე მის ბოლოში. დაზიანებისა და გაუმართაობის დროს მისი აღდგენის მეტი შანსია ასე.
- დანაყოფების UTF აღნიშვნის ველი. ასე, რომ დანაყოფს შეიძლება დავარქვათ სახელი ქართულ ენაზეც.
- EFI/UEFI firmware ნაგულისხმევი მნიშვნელობით იყენებს GPT-ს.

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

Ressource : https://en.wikipedia.org/wiki/Comparison_of_file_systems

ფაილური სისტემა	ფაილის დასახელების მაქსიმუმი სიგრძე	დასახელებაში ნებადართული ასო-ნიშნები	ფაილის სრული გზის მაქსიმალური სიგრძე	ფაილის მაქსიმალური ზომა	დისკის მაქსიმალური ზომა
FAT32	255 UTF-16 სიმბოლო	ნებისმიერი Unicode NUL-ის გარდა	შეზღუდვის გარეშე	4 GiB	512 MiB - 8 TiB
NTFS	255 სიმბოლო	ნებისმიერი Unicode NUL-ისა და /-ის გარდა	32,767 Unicode სიმბოლო. გზის სათითაო კომპონენტი 255 სიმბოლომდე	16 EiB	16 EiB
Ext4	255 ბაიტი	ნებისმიერი ბაიტი NUL-ისა და /-ის გარდა	შეზღუდვის გარეშე	16 GiB - 16 TiB	1 EiB
XFS	255 ბაიტი	ნებისმიერი ბაიტი NUL-ისა და /-ის გარდა	შეზღუდვის გარეშე	8 EiB	8 EiB

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

`fdisk, fdisk -l /dev/sda {m-დახმარება;d-დანაყოფის წაშლა;n-ახალი დანაყოფის შექმნა;p-დანაყოფების ცხრილის გამოტანა;w-ჩაწერა} ;`
`parted, gparted, parted -l, parted /dev/sda1 'unit MB print'; ერთეული={B,KiB,MiB,GiB,TiB,KB,MB,GB,TB,%,compact (human readable),chs}`

`blkid` – ბლოკური მოწყობილობების მახასიათებლების გამოტანა

`mkfs: mkfs -t <type de fichier> <partition>; mkswap`

`ext3: mke2fs -j, mkfs.ext3; ext4: mkfs.ext4 xfs: mkfs.xfs; Vfat: mkfs.vfat; XFS: mkfs.xfs`

`mkfs.ext3 /dev/sda1 = mkfs -t ext3 /dev/sda1 = mke2fs -j /dev/sda1 ;`
`mke2fs -c /dev/sda1` – ამოწმებს დაზიანებულ ბლოკებს (bad blocks) მოწყობილობაზე (read-only test); `badblocks -nsv /dev/sda1 ;`
`mke2fs -c -c /dev/sda1` – ამოწმებს დაზიანებულ ბლოკებს მოწყობილობაზე (read-write test; შლის მონაცემებს); `badblocks -wsv /dev/sda1 ;`
`mke2fs -F /dev/sda1` – მონტირებულ დანაყოფზე ქმნის ფაილურ სისტემას დაძალებით ;
`mkfs -t ext4 -v -N 1000 /dev/sda1` – ქმნის ფაილურ სისტემას ინოუდების მოცემული რაოდენობით

`tune2fs <partition>`

`tune2fs -l /dev/sda1` – გამოაქვს ფაილური სისტემის პარამეტრები`tune2fs -L /dev/sda1` – არქმევს სახელს ;
`tune2fs -c 8 -i 15 /dev/sda1` – ფაილური სისტემის შემოწმების განრიგის დაყენება: 8 მონტირების შემდეგ ან ბოლო შემოწმებიდან 15 დღის შემდეგ ;
`tune2fs -m 7 /dev/sda1` – რეზერვის ნაგულისხმევი მნიშვნელობის (5%) შეცვლა 7%-მდე ;
`tune2fs -j /dev/sda1` – ext2 დანაყოფის გადაკეთება ext3-ზე

`tune2fs -O extents,uninit_bg,dir_index /dev/sda1` – ext3 დანაყოფის გადაკეთება ext4-ზე

`fsck -t <type de fichier> <partition>`, `fsck.ext4`, `fsck.reiserfs`, `fsck.vfat`, `e2fsck`

`fsck -AR -y /dev/sda1` – ამოწმებს ყველა ფაილურ სისტემას (მოცემული `/etc/fstab`-ში) ერთი გაშვებით (`-A`) და root ფაილრი სისტემის გამოტოვებით (`-R`), `-y` = დასტური (yes) ყველა კითხვაზე

`dumpe2fs <partition>: dumpe2fs -h /dev/sda1` – სუპერბლოკის ინფორმაციის გამოტანა (superblock)

EXT4 დისკის განლაგება : {boot sector, block groupe1,block groupe2, . . . block groupeN}

EXT4 ბლოკის განლაგება : {superblock, group descriptor, reserved blocks, data block bitmap, inode bitmap, inode table, data blocks}

`chattr -/+/= attribut fichiers; lsattr fichiers`

ატრიბური={ a (დამატების რეჟიმი ჩაწერისას), A (« atime » არ იცვლება), i (წაშლა, შეცვლა, სახელის შეცვლა, მასზე ლინკის გაკეთება არ ხდება), e (აღნიშნავს, რომ ფაილი იყენებს extents-ს (დიდი ზომის ფაილების მონაცემთა ბლოკზე მიმართვა ბევრჯერ რომ არ მოხდეს, extent-ის შემოღებით მიჯრილი მონაცემების მხოლოდ საწყისი და ბოლო ბლოკის მისამართები ინახება. ეს საგრძნობლად ამაღლებს დიდი ზომის ფაილების კითხვის სიჩქარეს). chattr-ით მისი გამორთვა არ მოხდება}

როგორ დავიცვათ ფაილი: `chattr +i files`

როგორ დავიცვათ დირექტორია: `chattr -R +i folder`

მონაცემების დამატება არსებული მონაცემების შეუცვლელად ფაილში: `chattr +a files`

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

კვანძი (inode) არის მონაცემთა სტრუქტურა, რომელიც შეიცავს ფაილურ სისტემაში შენახული ფაილის შესახებ ინფორმაციას. თითოეულ კვანძს შეესაბამება ნომერი. კვანძი შეიცავს შემდეგ მონაცემებს :

- ფაილის ზომა ბაიტებში ; მოწყობილობის იდენტიფიკატორი, სადაც ეს ფიალი არის შენახული ; ფაილის მფლობელი / ფაილის მფლობელის ჯგუფი ; კვანძის ნომერი ფაილურ სისტემაში ; წვდომის უფლებები ;
- დროითი მახასიათებლები (timestamp) :
  - ctime (`stat` ან `ls -lc`) ;
  - mtime (`stat` ან `ls -l`) ;
  - atime (`stat` ან `ls -lu`) ;
- ბმულების რეოდენობა ამ ინოუდზე (Nlinks).

`stat file; ls -i file; find . -inum 12345 ; df -i` – კვანძების გამოყენებადობა პროცენტებში დანაყოფზე;

`debugfs -R 'stat /path/file' /dev/sdaN ; debugfs -R 'stat <file inode>' /dev/sdaN` – შექმნის დრო (crtime) ext4-ში

`/proc/filesystems` – გამოიტანს ბირთვის მიერ მხარდაჭერილი ფაილური სისტემების სიას.

`df` (disk free), `df -h` (human readable), `df -T` (გამოიტანს ფაილური სისტემის ტიპსაც)

`du` (disk usage), `du -h /dir` (დირექტორიებისა და ქვედირექტორიების ზომებს), `du -ah /dir` (ფაილების ზომებსაც), `du -sh /dir` (მხოლოდ ჯამური ზომა), `du -ah –exclude="*.txt" /dir` (ყველაფერს მოცემული გამოსახულების გამოტოვებით).

`/etc/mtab, /proc/mounts` (ამჟამად მონტირებული ფაილური სისტემები); `/etc/fstab` :

`device-spec` – მოწყობილობის სახელი, UUID, ან სხვა მახასიათებელი ;
`mount-point` – შესაბამისი დირექტორია; swap დანაყოფისთვის ეს ველი არის none ;
`fs-type` – ფაილური სისტემის ტიპი ;
`options` – ოფციები ;
`dump` – რამდენად ხშირად უნდა მოხდეს მისი სარეზერვო ასლის შენახვა dump პროგრამით. 0 ნიშნავს არასდროს (ანტომატურად). ძველია !;
`pass` – რიგითი ნომერი `fsck` პროგრამით შემოწმებისას.

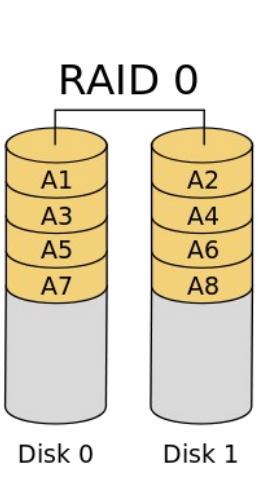
`mount` **ოპციები** = {`auto` (შეიძლება დამაუნტდეს `mount -a -თი`), `atime/noatime` (განაახლოს/არ atime ფაილურ სისტემაში), `exec/noexec` (პროგრამების გაშვება/არ გაშვების უფლება), `suid/nosuid` (set-user-identifier-ის მხარდაჭერა/არ), `user` (როგორც წესი, მხოლოდ ადმინისტრატორს შეუძლია ფაილური სისტემის მონტირება. `user` -ის მითითებით უკვე ყველა მომხმარებელს ექნება უფლება. მონტირების მოხსნაც თუ გვსურს მაშინ "`users`" უნდა მივუთითოთ "`user`"-ის ნაცვლად), `nouser, ro, rw, defaults` (`rw, suid, dev, exec, auto, nouser, async`)}

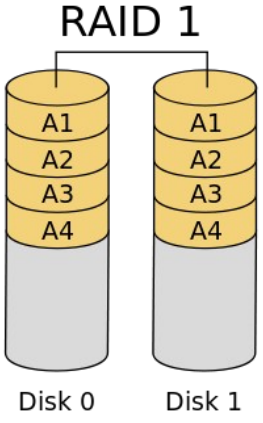
`mount` - ყველა მონტირებული სისტემის ნახვა ;
`u/mount -a` - /etc/fstab-ში ნახსენები ყველა ფაილური სისტემის მონტირება/მოხსნა ;
`mount -t iso9660 -o ro /dev/cdrom /dir` - CD/DVD-ROM-ის მონტირება ;
`mount -B /old_dir /new_dir` - ახალ დირექტორიაზე მიბმა ;
`mount -o remount,rw /dir` - ხელახლა მონტირება ;
`mount -t iso9660 -o loop file.iso /dir` - iso ფაილის მონტირება დირექტორიაზე ;
`umount /dir -l` - (Lazy unmount) მონტირების მოხსნა, როდესაც დისკზე ყველა ოპერაცია დამთავრდება.

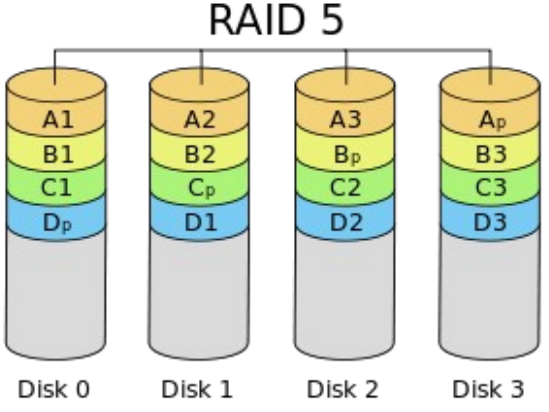
Ressource : https://www.prepressure.com/library/technology/raid

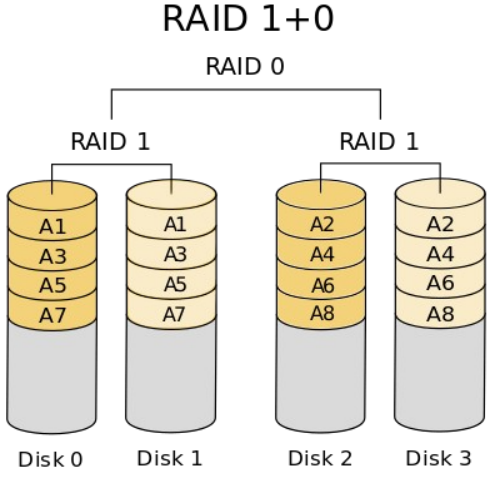
Redundant Array of Independent Disks.

RAID 0: In this level, a striped array of disks is implemented. The data is broken down into blocks and the blocks are distributed among disks. Each disk receives a block of data to write/read in parallel. It enhances the speed and performance of the storage device. There is no parity and backup in Level 0.
RAID 1: It uses mirroring techniques. When data is sent to a RAID controller, it sends a copy of data to all the disks in the array. RAID level 1 is also called mirroring and provides 100% redundancy in case of a failure.
RAID 5: It writes whole data blocks onto different disks, but the parity bits generated for data block stripe are distributed among all the data disks rather than storing them on a different dedicated disk.
RAID 6: It is an extension of level 5. In this level, two independent parities are generated and stored in distributed fashion among multiple disks. Two parities provide additional fault tolerance. This level requires at least four disk drives to implement RAID.
RAID 10: RAID 1+RAID 0. **Parity** computations are used in RAID drive arrays for fault tolerance by calculating the data in two drives and storing the results on a third. The parity is computed by XOR'ing a bit from drive 1 with a bit from drive 2 and storing the result on drive 3. After a failed drive is replaced, the RAID controller rebuilds the lost data from the other two drives. RAID systems often have a "hot" spare drive ready and waiting to replace a drive that fails.









Ressource : https://www.digitalocean.com/community/tutorials/how-to-create-raid-arrays-with-mdadm-on-ubuntu-16-04 ; Ressource : http://wiki.linuxservertech.com/index.php?action=artikel&cat=34&id=11&artlang=en; Ressource : https://www.howtoforge.com/software-raid1-grub-boot-debian-etch; Ressource : http://www.tecmint.com/manage-software-raid-devices-in-linux-with-mdadm/

`lsblk`; `lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT` (Block devices list)

`mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sdb /dev/sdc` (Create the Array)

`watch cat /proc/mdstat` (monitor the progress) ; `mkfs.ext4 /dev/md0` (Create and Mount the Filesystem) ; `df -h -x devtmpfs -x tmpfs` (Check space)

`mdadm --detail --scan | tee -a /etc/mdadm/mdadm.conf` (Save the Array Layout)

`update-initramfs -u` (Update the initramfs, or initial RAM file system, so that the array will be available during the early boot process)

`echo '/dev/md0 /mnt/md0 ext4 defaults,nofail,discard 0 0' | tee -a /etc/fstab` (Mount at boot)

`mdadm --add /dev/md0 /dev/sdc` (Replace a removed failed drive) ; `mdadm --detail /dev/md0` (Check RAID status)

`mdadm --examine /dev/sdc` (get RAID information on sdc if it is an array member)

`mdadm --query /dev/md0` (get information on a RAID array) ; `/usr/share/mdadm/mkconf --generate > /etc/mdadm/mdadm.conf` (Generate mdadm.conf)

`sfdisk -d /dev/sdb|sfdisk /dev/sdc; mdadm --zero-superblock /dev/sdc` (Copy the partition structure when replacing a failed drive and clean up disk /dev/sdc)

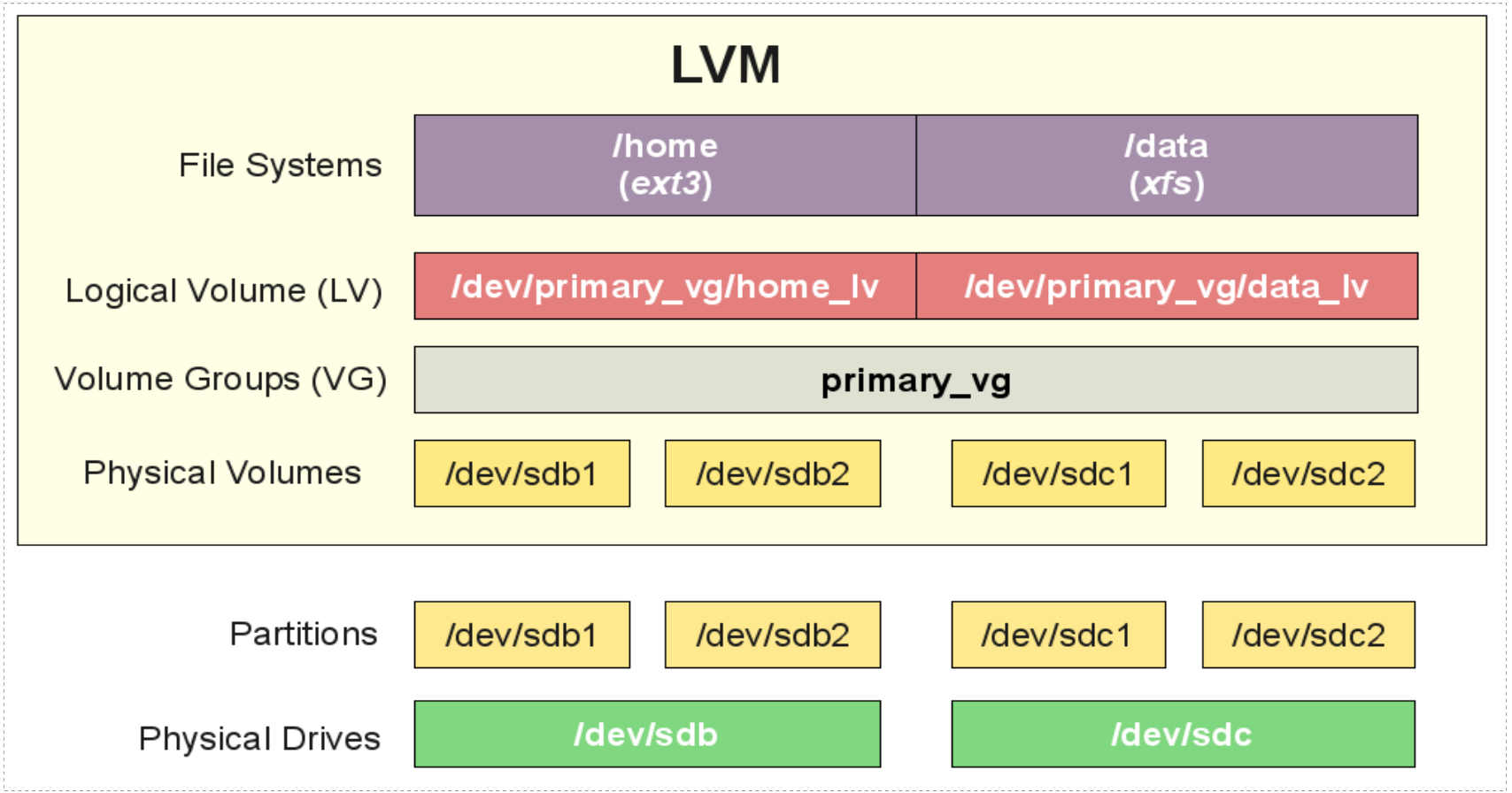
`umount /dev/md0; mdadm --stop /dev/md0; mdadm --remove /dev/md0` (Stop and remove the array)

`umount /dev/md0; mdadm --manage /dev/md0 --readonly; mount /dev/md0` (Marking an Raid array as ro)

`mdadm --manage /dev/md0 --fail /dev/sdc` (Mark /dev/sdc as failed drive)

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024



Ressource : https://www.howtoforge.com/linux_lvm

`apt-get install lvm2` – Install lvm tools

`fdisk /dev/sd[bc]` – Create a partition with Linux LVM type (8e) ; `fdisk -l` – Check partitions

`pvcreate /dev/sdb1 /dev/sdc1 /dev/sdb2 /dev/sdc2...` – Create a physical volume

`pvremove /dev/sdc2` – Remove sdd1 from physical volume; `pvdisplay; pvs` – Report information about PV

`vgcreate vg_name1 /dev/sdb1 /dev/sdc1 ...` – Create a volume group "vg_name1" with sdb1, sdc1 ...

`vgcreate vg_name2 /dev/sdb2 /dev/sdc2 ...` – Create a volume group "vg_name2" with sdb2, sdc2 ...

`vgdisplay; vgscan; vgs` – Report information about VG; `vgremove vg_name5` – Remove volume group "vg_name5"

`vgrename vg_name2 vg_name5; vgreduce vg_name1 /dev/sdc1` - Remove "/dev/sdc1" from "vg_name1"

`lvcreate --name lv1 --size 200M vg_name1` – Create a logical volume "lv1" in vg_name1

`lvrename vg_name1 lv1 lv2` – Rename lv1 with lv2 in vg_name1

`lvscan; lvdisplay; lvremove /dev/vg_name1/lv1; lvs` – Report information about LV

Enlarge lv1 from 200M to 1.2G - `lvextend -L1.2G /dev/vg_name1/lv1`; Then - `resize2fs /dev/vg_name1/lv1`

Shrink lv1 to 1G : First - `resize2fs /dev/vg_name1/lv1 1G`, Then - `lvreduce -L1G /dev/vg_name1/lv1`

```
mkfs.ext4 /dev/vg_name1/lv1; mkfs.xfs /dev/vg_name2/lv1 … ;
mount /dev/vg_name1/lv1 /dir1; mount /dev/vg_name2/lv1 /dir2 … ; df -h
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

Sytem V

- SysVinit (Debian 6 and earlier ; Ubuntu 9.04 and earlier ; CentOS 5 and earlier)
  - /etc/inittab
  - init N, telinit N, init 1 = init s, init q
- ჩატვირთვის დონეები :
  - /etc/init.d/* ;
  - /etc/rcN.d/[SK]N*;
  - service service_name start|stop|restart|reload|status
- Debian: update-rc.d
  - ex : update-rc.d apache2 start 20 2 3 4 5 . stop 80 0 1 6 .
- Red hat: chkconfig (--add, --del, --list) cmd
  - ex : chkconfig --level 2 programe on ; chkconfig --list

Ressource : https://www.digitalocean.com/community/tutorials/systemd-essentials-working-with-services-units-and-the-journal
Ressource : https://www.digitalocean.com/community/tutorials/understanding-systemd-units-and-unit-files
Ressource : https://www.certdepot.net/rhel7-get-started-systemd/

- Systemd (Debian 7 and Debian 8 ; Ubuntu 15.04 and newer ; CentOS 7) ჩატვირთვა უფრო სწრაფია, რადგან შედარებით ცოტა სკრიფტებს იყენებს და ცდილობს ამოცანები (units) პრალელურად გაუშვას. (Lennart Poettering)
- Units (.service, .target …)
- `/etc/systemd/` - Systemd-ს გლობალური კონფიგურაცია; `/lib/systemd/system/` - Service-ების კონფიგურაციის ფაილები ; `/etc/systemd/system/` - დამატებითი (Custom) service-ების კონფიგურაციის ფაილები.
- `systemctl --version` – Systemd-ს ვერსია.
- `systemd-analyze [blame|critical-chain]` – ჩატვირთვის პროცესის ხანგრძლივობა
- `systemctl start|stop|reload|status|enable|disable|is-enabled|is-active UNIT_name`
- `systemctl isolate poweroff.target|rescue.target|multi-user|graphical.target|reboot.target` – ჩატვირთვის დონის შეცვლა (target არის ჯგუფური მექანიზმი, რომელიც ერთდოულად ტვირთავს სხვადასხვა სერვისებს. კონკრეტულ target-თან ასოცირებული სერვისები ინახება target დასახელების დირექტორიაში სუფიქსით".wants". გარკვეულ შემთხვევებში targets ცვლის ჩატვირთვის დონეს, ტუმცა ის უფრო ზოგადია)
- `systemctl poweroff|reboot|rescue`
- `systemctl daemon-relaod`
- `systemctl list-units [--all|--type=]` - ანახებს ყველა ჩატვირთულ unit-ებს მისი მდგომარეობის მიუხედავად.
- `systemctl list-unit-files` – გამოაქვს სისტემაში დაინსტალირებული ყველა unit.

Ressource: https://www.digitalocean.com/community/tutorials/systemd-essentials-working-with-services-units-and-the-journal
Ressource: https://www.digitalocean.com/community/tutorials/how-to-use-systemctl-to-manage-systemd-services-and-units
Ressource: http://www.raspberrypi-spy.co.uk/2015/10/how-to-autorun-a-python-script-on-boot-using-systemd/

- `systemctl get-default`
- `systemctl set-default multi-user.target; systemctl set-default graphical.target`
- `systemctl list-dependencies multi-user.target; systemctl list-dependencies graphical.target`
- `systemctl cat|show|edit`(systemd version 218)`|mask unit_file`

Control Groups: Systemd პროცესებს აერთიანებს საკონტროლო ჯგუფებში (control group). მაგალითად, apache ვებ სერვისის მიერ გაშვებული პროცესები გაერთიანებული იქნება ერთ საკონტროლო ჯგუფში, მათ შორის CGI სკრიფტები.

- `systemcd-cgls` – ანახებს საკონტროლო ჯგუფების სრულ იერარქიას.
- `systemcd-cgtop` – პროცესორის, მეხსიერების, დისკის შეტან/გამოტანის დატვირთვა
- საკონტროლო ჯგუფების მიერ.
- `systemctl kill httpd` - apache-თან ასოცირებული ყველა პროცესის მოკვლა.
- `systemctl -H root@IP[:httpsrv] status httpd.service` – დისტანციური ინფორმაცია.

სკრიფტის გაშვება სისტემის ჩატვირთვისას :

- საკუთარი სკრიფტის შექმნა : `/full_path/myscript.sh`
- Unit ფაილის შექმნა : `/etc/systemd/system/myscript.service`
- Systemd-ს დაკონფიგურირება: `systemctl daemon-reload;systemctl enable myscript.service`
- სერვისის სტატუსი ჩატვირთვის შემდეგ : `systemctl status myscript.service`

/etc/systemd/system/myscript.service:

```
[Unit]
Description=My Script Service

[Service]
Type=simple
ExecStart=/PATH/myscript.sh

[Install]
WantedBy=default.target
```

Ressource : https://www.digitalocean.com/community/tutorials/how-to-use-journalctl-to-view-and-manipulate-systemd-logs

Systemd, დამატებით, მართავს სისტემის ჟურნალებს (system event log) Journald-ს საშუალებით.
rsyslogd დემონი, რომელიც ნაგულისხმევი მნიშვნელობიტ ჯერ კიდევ წარმოდგენილია, აღარ არის აუცილებელი !!!

```
journalctl; journalctl --utc
```

გაფილტვრა დროის მიხედვით :

- `journalctl --b` (ანახებს Log-ებს მიმდინარეჩატვირთვიდან)
- `journalctl --list-boots` (წინა ჩატვირთვები)
- `journalctl -b -1`
- `journalctl --since "YYYY-MM-DD HH:MM:SS"`
- `journalctl --since "YYYY-MM-DD HH:MM:SS" --until "YYYY-MM-DD HH:MM:SS"`
- `journalctl --since yesterday`
- `journalctl --since 09:00 --until "1 hour ago"`

ნაგულისხმევი მნიშვნელობით, Journald-ს ჟურნალები ინახება `/run/log/journal/`-ში და გადატვირთვის შემდეგ იკარგება.

Journald log-ების მუდმივად შესანახად უნდა აკრიფოთ:

- `mkdir /var/log/journal; systemd-tmpfiles --create --prefix /var/log/journal`
- `echo "SystemMaxUse=50M" >> /etc/systemd/journald.conf; systemctl restart systemd-journald`

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

Systemd journald

Ressource: https://www.digitalocean.com/community/tutorials/how-to-use-journalctl-to-view-and-manipulate-systemd-logs
Ressource: https://wiki.debian.org/systemd

- გაფილტვრა შემდეგი კრიტერიუმების მიხედვით : unit, process, UID, GID, path, priority, kernel message.
  - `journalctl -u unit.service [--since today] ; journalctl -u unit1.service -u unit2.service [--since today]`
  - `journalctl _PID=1234 ; man systemd.journal-fields`
  - `id -u www-data  =>  1000 ; jourmannalctl _UID=1000 --since today`
  - `journalctl /usr/sbin/sshd ; journalctl -k` (kernel messages)
  - `journalctl -p err ;` Priority: `{0:emerg, 1:alert, 2:crit, 3:err, 4:warning, 5:notice, 6:info, 7:debug}`
  - `journalctl -o {json, json-pretty, sort …} ; journalctl -n [N]` (most recent 10 entries)
  - `journalctl -f` (following logs) ; `journalctl --disk-usage` (following logs)
- გამართვა (Debuging). ზოგჯერ საჭიროა გამოვიკვლიოთ ხოლმე ის მიზეზები, რის გამოც სისტემა დაეკიდა/გაიჭედა : გამოსავალი ასეთია :
  - ბირთვის ბრძანებათა ხაზიდან (ე.წ. "cmdline" ან "grub line") "quiet" გამოსახულების ამოშლა; ბირთვის პარამეტრები/ოფციების შემოწმება: `cat /proc/cmdline`
  - მეტი ინფორმაციის გამოტანა (verbosity): უნდა დაემატოს "systemd.log_target=kmsg systemd.log_level=debug"
  - მეტი ინფორმაციის გამოტანა (verbosity) `/etc/systemd/system.conf` ფაილის გამოყენებით: `LogLevel=debug; LogTarget=syslog` (ან `kmsg`)
  - ავარიული შელის ჩატვირთვა: დაუმატეთ systemd.unit=rescue.target ან უბრალოდ 1 ბირთვის ბრძანებათა ხაზს.
  - debug shell-ის გააქტიურება: უნდა გაუშვათ `systemctl enable debug-shell.service`. (შეგიძლიათ ეს ავარიული რეჟიმის ჩატვირთვის შემდეგ გაუშვათ) ასე გაეშვება root shell TTY 9 ტერმინალზე.

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ჟურნალიზაცია Syslogd

/var/log, /var/log/messages

[r]syslogd, /etc/[r]syslog.conf

groupe1.priority1 ; groupe2.priorité2 ; .../var/log/fichier_de_log

ჯგუფი : {auth, authpriv, cron, daemon, kern, lpr, mail, mark, news, syslog, user}

პრიორიტეტი : {emerg,alert,crit,err,warning,notice,info,debug,*,none}

ჟურნალების სერვერი :

groupe1.priorité1 ; groupe2.priorité2 ; @IP

/etc/init.d/syslogd; RSYSLOGD_OPTIONS="-r"

logger « message », logrotate, /etc/logrotate.conf

Hostname

Ressource: http://www.cyberciti.biz/faq/rhel-redhat-centos-7-change-hostname-command/
Ressource: https://www.cyberciti.biz/faq/rhel-redhat-centos-7-change-hostname-command/

- Core networking (debian):
  - Update `/etc/hostname, /etc/hosts`
  - `invoke-rc.d hostname.sh start ; invoke-rc.d networking force-reload`
  - `invoke-rc.d network-manager force-reload`
- Systemd: არსებობს კომპიუტერის 3 სახის დასახელება: static, pretty, and transient. static არის ტრადიციული დასახელება. მისი არჩევა მომხმარებელს შეუძლია და ის ინახება /etc/hostname ფაილში. transient არის ბირთვის მიერ მხარდაჭერილი დინამიკური დასახელება. ნაგულისხმევი მნიშვნელობა static-დან აიღება და არის localhost. ის შეიძლება შიცვალოს DHCP-სა და DNS-ის მიერ. pretty არის მომხმარებლის მიერ შერჩეული თავისუფალი UTF8 ფორმატის დასახელება.
  - `hostnamectl or hostnamectl status`
  - `hostnamectl set-hostname achiko`
  - `hostnamectl set-hostname "achiko.ifg.tsu.ge" --static`
  - `hostnamectl set-hostname "achiko" --trasient`
  - `hostnamectl set-hostname "Achiko’s PC in the Filière" --pretty`
  - `hostnamectl set-hostname ""`
  - `hostnamectl set-hostname "" --pretty|static|trasient`

პროცესორი (Central Processor Unit ან CPU)

მეხსიერება : « ცოცხალი » მეხსიერება (Random access memory ან RAM) ; « მკვდარი » მეხსიერება (Read Only Memory ან ROM)

სალტე (bus) ; დისკები ;

BIOS (Basic Input Output System), POST (Power-On Self Test)

UEFI (Unified Extensible Firmware Interface)

BIOS ტვირთავს დისკის საწყის სექტორს და უშვებს მას. ეს ჩამტვირთავი სექტორი (boot sector), თავის მხრივ, უშვებს დამატებით კოდს. BIOS მეტ-ნაკლებად შეზღუდული სისტემაა, შეზღუდული სივრცით, რადგან BIOS უშვებს 16-ბიტიან კოდს, მაშინ როდესაც, თანამედროვე კომპიუტერები იყენებენ 32 და 64 ბიტიან CPU-ს. EFI (ან UEFI, რომელიც არის უბრალოდ EFI 2.x) ჩართვისას ტვირთავს EFI პროგრამის ფაილებს (.efi გაფართოებით) დისკის დანაყოფიდან. ეს დანაყოფი ცნობილია როგორც ESP (EFI System Partition). ამ EFI ჩამტვირთავი პროგრამების უპირატესობა უბრალოდ EFI ჩატვირთვის სერვისთან შედარებით ისაა, რომ მას შეუძლია ფაილები წაიკითხოს პირდაპირ დისკიდან.

UEFI აგრეთვე აქვს უსაფრთხო ჩატვირთვის (Secure Boot) მხარდაჭერა. ყოველივე ეს ხორციელდება ციფრული ხელმოწერის გამოყენებით ჩამტვირთავ პროგრამებში, რომლებიც, თავის მხრივ, მას იყენებენ ჩასატვირთი ბირთვის იდენთიფიცირებისათვის და ა.შ. ასე, ბოროტმოქმედისათვის ჩატვირთვის პროცესში ჩართულობა რთულდება, რაც უფრო მეტ უსაფრთხოება უწყობს ხელს.

როგორ ჩაიტვირთა კომპიუტერი? UEFI-ით თუ BIOS-ით ?

თუ /sys/firmware/efi არსებობს, მაშინ ჩატვირთვა EFI რეჟიმში მოხდა.

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ბირთვის 2.6 ვერსიიდან დაწყებული პერიფერიული მოწყობილობების შესახებ უმეტესი ინფორმაცია გადავიდა `/sys`-ში. Sysfs ფაილური სისტემა, არამხოლოდ თავად მოწყობილობების შესახებ გვეუბნება ინფორმაციას, არამედ ის გამოიყენება ბირთვის გარკვეული ფუნქციონალს კონფიგურაციისთვისას.

დემონი udevd უყურებს ბირთვის შეტყობინებებს მოწყობილობების მდგომარეობების ცვლილების შესახებ. udevd იყენებსა ამ და, დამატებით, /sys ფაილური სისტემიდან აღებულ ინფორმაციას, რათა განახორციელოს შესაბამისი ქმედება მოწყობილობის აღსაქმელად მომხმარებლის დონეზე – ჩატვირთოს მოდული ან დრაივერი ან firmware და შექმნას `/dev`-ში შესაბამისი სპეციალური ფაილი.

D-Bus არის პროცესებს შორის კომუნიკაციის სისტემა. ის იყენებს sysfs-ს და სისტემაში ავრცელებს შესაბამის ცნობებს – მაგ : « დაემატა ახალი მოწყობილობა » და ა.შ.

დემონი hald დაკავშირებულია D-Bus-თან, რათა შესთავაზოს აპლიკაციების პროგრამირების ინტერფეისი (API), იმისთვის, რომ პროგრამებს შეეძლოთ მოწყობილობების მართვა და მათთან ოპერირება. სწორედ ეს API ქმნის მოწყობილობის აბსტრაქტულ ფენას (Hardware Abstraction Layer - HAL).

/proc/interrupts, /proc/ioports

წყვეტა (IRQ - Interrupt Request). მოწყობილობა ან პროგრამა იყენებს წყვეტას პროცესორთან ურთიერთობისას. რათა მას გარკვეული ამოცანა გადასცეს დასათვლელად. როგორც კი პროცესორი მიიღებს წყვეტას, ის დორებით ჩერდება (აჩერებს მიმდინარე გამოთვლას) და უშვებს სპეციალურ პროგრამას Interrupt Handler ან ISR (Interrupt Service Routine). ეს სერვისი არის გარკვეულ ცხრილში (Interrupt Vector table), რომელიც თავის მხრივ მდებარეობს მეხსიერების ფიქსირებულ მისამართზე. როგორც კი ეს წყვეტა დამუშავდება CPU აგრძელებს შეჩერებულ გამოთვლას.

ზოგადად CPU-სთან მიმართებაში მოწყობილობებს ორი საშუალება გააჩნიათ: წყვეტაზე (Interrupt based) ან არჩევანზე (Polling based) დაფუძნებული. **ლინუქსის ყველა სისტემა წყვეტაზეა დაფუძნებული.**

IRQ 0 — system timer (cannot be changed); IRQ 1 — keyboard controller (cannot be changed);

IRQ 3 — serial port controller for serial port 2 (io-port=0x2F8) (/dev/ttyS1) ; shared with serial port 4, if present;

IRQ 4 — serial port controller for serial port 1 (io-port=0x3F8) (/dev/ttyS0) ; shared with serial port 3, if present;

IRQ 5 — parallel port 2 and 3 or sound card; IRQ 6 — floppy disk controller;

IRQ 7 — parallel port 1. It is used for printers or for any parallel port if a printer is not present.

IRQ 8 — rtc(real time clock=hwclock). The RTC is almost never used by the Linux kernel. Instead, the kernel uses the software clock time, timestamps on files, etc. However, at boot time the kernel initializes its software clock by reading the RTC.

IRQ 12 — mouse; IRQ 14 — primary IDE/SATA controller; IRQ 15 — secondary IDE/SATA controller;

Irqbalance პროგრამა წყვეტებს ანაწილებს პროცესორის ბირთვებზე, რაც ზრდის სისტემის წარმადობას. Irqbalance-ის მიზანია ბალანსი დაიჭიროს ოპტიმალურ წარმადობასა და ენერგიის დაძოგვას შორის.

```
apt-get install irqbalance ; systemctl start irqbalance
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

`dmesg, /proc/cpuinfo, /proc/meminfo, /proc/bus/pci, lspci [-v[vv]] [-t] lshw ; lsusb, lsusb -t ; udevadm monitor ;`
`udevadm info (მაგ: udevadm info -p (--path=) /sys/class/net/eth0/ -a)`

პერიფერიული მოწყობილობების შესახებ ინფორმაცია : `dmidecode -t (--type) N`

USB (Universal Serial Bus) - Hot Pluggable :

USB-1 : 1,5 Mbps - 12 Mbps ; USB-2 : 480 Mbps. USB-3 : 3 – 5Gbps

SSD (Solid State Drivers) use SAS, SATA or PCI-Express interfaces.

SATA (Serial Ata, SATA III -2009) : 6 Gbps (read speed), 120 Mbps (write speed),

SAS (Serial Attached SCSI (Small Computer System Interface) , SAS-3) : 12 Gbps (read speed)

PCIe (Peripheral Component Interconnect Express) : 985 Mbps (read speed PCIe Gen 3 lane)

Server throughput (write speed) :
`hdparm -W1 /dev/sda` (cache activated) ; `hdparm -W0 /dev/sda` (cache disactivated)

```
dd if=/dev/zero of=/tmp/test.img bs=1G count=1 oflag=dsync [direct]
```
(oflag=`dsync` : Use synchronized I/O for data. This option get rid of caching and gives you good and accurate results ; oflag=`direct` : Use direct I/O for data)

Server latency (write speed) : `dd if=/dev/zero of=/tmp/test2.img bs=512 count=1000 oflag=direct`

Server read speed :

```
echo 3 | sudo tee /proc/sys/vm/drop_caches
```
(To get accurate read test data, first discard caches before testing)

```
time dd  if=/path/to/bigfile of=/dev/null bs=8k
```

```
cat /dev/sda | pv -r > /dev/null
```

Buffered disk(-t) and cache(-T) read speed : `hdparm -tT /dev/sda`

How to copy CD/DVD into a file ?

```
cat /dev/sr0 >image.iso; cat </dev/sr0 >image.iso; tee </dev/sr0 >image.iso; dd </dev/sr0 >image.iso
dd if=/dev/cdrom of=image.iso; pv </dev/sr0 >image.iso; cp /dev/sr0 image.iso; tail -c +1 /dev/sr0 >image.iso
```

RAM speed (in MHz) : `dmidecode --type 17  | grep -i speed`

თუ შიდა მოწყობილობების კონფიგურაციისას საჭიროა წყვეტისა და შეტან-გამოტანის მისამართის ცოდნა, გარე მოწყობილობების დროს, როგორიც მოდემია, საჭიროა ვიცოდეთ ის პორტი, რომელზეც არის მიერთებული მოწყობილობა. გარე მოდემი ერთდება მიმდევრობით ინტერფეისზე (პორტზე), რომლის შესაბამისი ფაილიც არის **/dev/ttyS0** (პირველი) და **/dev/ttyS1** (მეორე).

ბრძანება `setserial` ასკნირებს მიმვდევრობით პორტებს და ეკრანზე გამოაქვს ის რესურსები, რაც მასზე მიერთებულ მოწყობილიობებს აქვთ მინიჭებული. ამ ბრძანებით აგრეთვე შესაძლებელია ჩვენით მივუთითოთ რესურსები (წყვეტა და შეტან-გამოტანის მისამართი) შიდა მოდემისთვის, აგრეთვე განვსაზღვროთ სიჩქარეც :

```
setserial /dev/ttyS3 port 0xe800 irq 11 autoconfig
```

`autoconfig` ოფცია ავტომატურად დაინახავს ამ მოწყოვბილობას (UART - Universal Asynchronous Receiver Transmitter).

არსებობს 3 ძირითადი UART-ის ტიპი:

8250 – პორტის სიჩქარეა 19200 bps მაქსიმუმ.

16450 - სიჩქარე 38400-დან 57600 bps-მდე.

16550 – სიჩქარე 115 kbps (მაქსიმუმ). ის მიიღწევა 2 FIFO ბუფერის წყალობით (16 byte შიგნით ; 16 byte გარეთ).

```
minicom ; setserial speed option

option={spd_hi(56ko),spd_vhi(115ko),spd_dhi(230ko),spd_warp(460ko),spd_norma(38.4ko),spd_cust}
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ლოკალიზაცია
ინტერნაციონალიზაცია
დრო

`locale,locale -a` (დაყენებული ლოკალიზაციები); ნაგულისხმევი მნიშ.:`/etc/default/locale`(debian),`/etc/sysonfig/i18n`(red hat)

`dpkg-reconfigure locales ; man man` (ინგლისურად); `export LANG=fr_FR.utf8 ; man man` (ფრანგულად)

`localectl [status]; localectl set-locale LANG=en_US.utf8; setxkbmap -layout ge (us, fr)`

მუდმივად : `/etc/sysconfig/keyboard` (red hat), `/etc/default/keyboard` (debian) (`XKBLAYOUT="us" by XKBLAYOUT="fr"`) ; `reboot`.

```
timedatectl [status] ;
timedatectl list-timezones ; timedatectl set-timezone "Asia/Tbilisi" ;timedatectl set-timezone UTC ;
timedatectl set-time YYYY-MM-DD; timedatectl set-time HH:MM:SS; timedatectl set-time 'HH:MM:SS YYYY-MM-DD'
```

`/etc/localtime =>/usr/share/zoneinfo/UTC..`, `/etc/timezone` (დროითი ზონა, სასაათო სარტყელი)

`/usr/share/zoneinfo/` - ყველა დროითი ზონის სია ; `tzselect ; TZ env varible (export TZ=UTC+2) ; date +%Z`

`date -u` (universal time = UTC=GMT=greenwich mean time) ; `date MMDDhhmmYY.SS`

`hwclock (--show=-r ; -systohc=-w` (hc-ს გასწორება sys-ზე) `;-hctosys=-s` (sys-ის გასწორება hw-ზე) `; -adjust=-a ; --utc=-u)`
სისტემის დროის საწყისი მნიშვნელობა აიღება მოწყობილობის დროიდან. `hwclock` იყენებს `/etc/adjtime` ფაილს, რომ აკონტროლოს დაზუსტება.

`timedatectl set-local-rtc 1` (hw-ს დაყენება ლოკალურ დროის სარტყელზე);`timedatectl set-local-rtc 0` (hw-ს დაყენება UTC-ზე)

`timedatectl set-ntp true [false]` (დროის ავტომატური სინქრონიზაცია NTP სერვერთან)

Ressource : http://www.gnu.org/software/grub/manual/grub.html

GRUB v2 - /boot/grub/grub.cfg (DO NOT MODIFY) ; update-grub ; GRUB-legacy - /boot/grub/menu.lst ; GRUB-legacy დანაყოფების ათვლა იწყება 0-დან, GRUB2 - აქ 1-დან ; grub-legacy-ის (hd0,0) grub2-სთვის იქნება (hd0,1) ; GRUB არ არჩევს IDE და SCSI/SATA დისკებს

GRUB 2 უკეთესად ეძებს ფაილებს. მას ესმის ფაილების label და UUID (Universally Unique Identifiers)

GRUB 2 -ს შეუძლია წაიკითხოს ფაილები პირდაპირ LVM და RAID მოწყობილობებიდან აქვს რამდენიმე ფაილური სისტემის მხარდაჭერა

GRUB ძირითადად დისკიდან ტვირთავს ფაილებს, თუმცა ქსელიდანაც შეუძლია ჩატვირთვა TFTP პროტოკოლით.

```
/etc/default/grub ; /etc/grub.d/* ; grub-install /dev/sda
```

პაროლით დაცვა :

```
grub—mkpassd-pbkdf2
/etc/grub.d/40_custom
set superusers="username"
password_pbkdf2 username output_of_grub-mkpasswd-pbkdf2
```

N.B. უნდა დაემატოს "--unrestricted" მენიუს ჩანაწერში. ანუ, **/etc/grub.d/10_linux**-ში :CLASS="--class gnu-linux --class gnu --class os --unrestricted"

ფონის სურათის შეცვლა : /etc/default/grub - GRUB_BACKGROUND="/path_to_image"

Ressource : http://www.gnu.org/software/grub/manual/grub.html

GRUB-ის ბრძანებები (press c in grub2 menu. grub2-ის მენიუში შესვლა ჩატვირთვის დროს.)

`ls` – გამოიტანს დისკებისა და დანაყოფების სიას

`ls (hd0,1)` – გამოიტანს კონკრებულად ამ დანაყოფის დეტალებს (ტიპი,ზომა ...)

`ls (hd0,1)/` – გამოიტანს ამ დანაყოფზე ფაილების სიას (აქედანაც მივხვდებით რომ ეს არის root დანაყოფი) + ჩასატვირთი ბითვის სახელს.

`linux (hd0,1)/vmlinuz root=/dev/sda1` – ამ ბირთვის ჩატვირთვა. ეს დასახელება უნდა გამოიტანოს ls (hd0,1)/ ბრძანებამ

`initrd (hd0,1)/initrd.img – initrd.img` ფაილის ჩატვირთვა. ეს დასახელება უნდა გამოიტანოს ls (hd0,1)/ ბრძანებამ

`boot` – სისტემის ჩატვირთვა

grub2-ის მენიუდან ჩასატვირთი ბირთვის პარამეტრების შეცვლა (e ღილაკზე დაჭერა).

X server

Ressource: ftp://www.x.org/pub/X11R6.8.2/doc/xorg.conf.5.html

X-Window, MIT (Massachusetts Institute of Technology) , პროექტი ' athena'. **Xorg :** *freedesktop.org* საზოგადოება.

```
Xorg -configure ; /etc/X11/xorg.conf
```

Sections {`InputDevice` (Input device description), `Files` (File pathnames), `Device` (Graphics device description), `Monitor`(resolution), `Screen` (Device+Monitor), `ServerLayout`(Screen+InputDevices), `Module` (Dynamic module loading)}

Display Managers : GDM, KDM, MDM (Mint Display Manager), SliM(Simple Login Manager), SDDM (Simple Desktop Display Manager), LXDM, LightDM.

Window Mangers : Gnome, KDE, XFCE, Enlightenment, LXDE, Window Maker, Fluxbox, AmiWM, Sawfish, IceWM.

```
/etc/X11/default-display-manager ; DISPLAY=hostname:displayname.screenname; startx -- :N
```

How to start the second display on the local machine: `startx -- :1`

How to connect to the remote X server via xdmcp protocol :

On server :`/etc/gdm3/daemon.conf ; [xdmcp] => Enable=true ; Port=177 ; [MaxSessions=N] ;`

On client : `X :N -query IP ;  X :N -broadcast` (for automatic color management change: `auth_admin` with `yes` in `/usr/share/polkit-1/actions/org.freedesktop.color.policy`)

```
Xnest -query IP -geometry 1280x1024 :N
```

```
Xephyr -query IP -screen 1280x1024 -br -reset -terminate :N
```

```
xhost +/- IP ; dpkg-reconfigure x11-common (debian) ; dpkg-reconfigure gdm3
```

How to run a remote host application on your local display using ssh : `ssh -X remote_host "GUI_command"`

How to run a remote host application on the remote display using ssh : `ssh -X remote_host ; export DISPLAY=:0 ; GUI_command`

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

VNC (Virtual Network Computing) ; RFB (remote framebuffer) პროტოკოლი.

სერვერი : `vino; vnc4server ; tightvncserver` (read-only view also) `; $HOME/.vnc/xstartup`

`Vnc4server [-kill] :N ; tightvncpasswd`

კლიენტი : `[xg]vncviewer server_IP:N`

როგორ გავუშვათ vnc vino სერვერი დისტანციურად ssh-ით (GNOME-ში):

1) `ssh user@remote_host`

2) თუ მომხმარებელი უკვე შესულია, მაშინ `DISPLAY` იქნება `":0"`, თუ არა და ჯერ უნდა გავუშვათ `startx` (`#(debian8)dpkg-reconfigure x11-common, #(debian9)dpkg-reconfigure xserver-xorg-legacy; /etc/X11/Xwrapper.config: allowed_users=anybody; needs_root_rights=yes`), მაშინ `DISPLAY` იქნება `":1"`; ბოლოს კი ვუშვებთ შემდეგ ბრძანებებს :

- `DISPLAY=:0[1] gsettings set org.gnome.Vino require-encryption false`
- `DISPLAY=:0[1] gsettings set org.gnome.Vino prompt-enabled false`
- `DISPLAY=:0[1] gsettings set org.gnome.Vino view-only false`
- `DISPLAY=:0[1] gsettings set org.gnome.Vino vnc-password $(echo test |base64)`
  (If password is not set `(settings => sharing => screen sharing)` than no password is needed, otherwise it will overwrite the old password with the new one)
- `DISPLAY=:0[1] /usr/lib/vino/vino-server&`
- `DISPLAY=:0[1] gsettings list-keys org.gnome.Vino` – vino-ს პარამეტრების სია.

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ინტერპრეტატორები:

Bourne Shell ტიპის :
sh (Bourne Shell, AT&T Bell Labs)
ksh (KornShell : written by David Korn in the 1980s
zsh (Zsh is another shell which has similarities to ba
bash (Bourne Again Shell, Brian Fox, 19 GNU Projec

C shell ტიპის :
csh (Berkely Unix C shell)
tcsh (TENEX/TOPS C shell is a derivative of csh)

სკრიპტი:

```
nom_script.sh; sh nom_script.sh

echo " ls -l " > script

chmod +x script

#!/bin/bash [sh, csh ...]

./script
```

კომპილატორი	ინტერპრეტატორი
შესასვლელზე იღებს მთლიან პროგრამას	შესასვლელზე იღებს პროგრამის სათითაო ინსტრუქციას
გენერირდება შუამავალი კოდი	არ გენერირდება შუამავალი კოდი
ინსტრუქციები უფრო სწრაფად სრულდება	ინსტრუქციები უფრო ნელა სრულდება
მეტ მეხსიერებას იყენებს (მას შემდეგ, რაც შუამავალი კოდი დაგენერირდება)	ნაკლებ მეხსიერებას იყენებს
პროგრამა არ საჭიროებს კომპილაციას ყოველ ჯერზე გაშვებისას	პროგრამა საჭიროებს კოდის მაღალი დონიდან დაბალზე გადაყვანას ყოველ ჯერზე გაშვებისას
შეცდომები ეკრანზე გამოდის პროგრამის მხოლოდ სრულად შემოწმების შემდეგ	შეცდომები ეკრანზე გამოდის პროგრამის სათითაო ინსტრუქციის ინტერპრეტირებისას

```
test expression

[ expression ]

[[ expression ]]
```

პირობა = { expression (ჭეშმარიტი/მცდარი), ბრძანება }

ლოგიკური ოპერატორები: && - "და", || - "ან", ! - "არა"

არითმეტიკული ოპერატორები: -lt, -gt, -eq, -le, -ge, -ne

ფაილების შემოწმება: -e, -b, -c, -p, -f, -d, -h(-L), -r, -w, -x, -u, -g, -k, -O, -G, -s
(ფაილის არსებობს არანულოვანი ზომით)

```
fichier1 -nt fichier2, fichier1 -ot fichier2, hlien1 -ef hlien2
```

ცვლადების შემოწმება: -n $x, -z $y, $x =[=] $y, $x != $y

გადაბმა: condition1 -a condition2, condition1 -o condition2

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

Bash
Test, [, [[

Feature	ახალი test [[	ძველი test [	მაგალითი
string comparison	>	\> (*)	`[[ b > a ]] && echo "a does not come after b"`
	<	\< (*)	`[[ az < za ]] && echo "az comes before za"`
	= (or ==)	=	`[[ a = a ]] && echo "a equals a"`
	!=	!=	`[[ a != b ]] && echo "a is not equal to b"`
integer comparison	-gt	-gt	`[[ 5 -gt 10 ]] || echo "5 is not bigger than 10"`
	-lt	-lt	`[[ 8 -lt 9 ]] && echo "8 is less than 9"`
	-ge	-ge	`[[ 3 -ge 3 ]] && echo "3 is greater than or equal to 3"`
	-le	-le	`[[ 3 -le 8 ]] && echo "3 is less than or equal to 8"`
	-eq	-eq	`[[ 5 -eq 05 ]] && echo "5 equals 05"`
	-ne	-ne	`[[ 6 -ne 20 ]] && echo "6 is not equal to 20"`
conditional evaluation	&&	-a (**)	`[[ -n $var && -f $var ]] && echo "$var is a file"`
	||	-o (**)	`[[ -b $var || -c $var ]] && echo "$var is a device"`
expression grouping	(...)	\(... \) (**)	`[[ $var = img* && ($var = *.png || $var = *.jpg) ]] && echo "$var starts with img and ends with .jpg or .png"`
Pattern matching	= (or ==)	(not available)	`[[ $name = a* ]] || echo "name does not start with an 'a'"`
Reg. Expression matching	=~	(not available)	`[[ $(date) =~ ^Fri\ ...\ 13 ]] && echo "It's Friday the 13th!"`

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024


```
if [ condition ]
then
   commande1
   commande2
   ...
fi
```

```
if [ condition ]
then
   commande1
   commande2
   ...
else
   commande
fi
```

```
if [ condition1 ]
then
   commande1...
elif [ conditiion2 ]
   commande2...
   ...
else
   commande
fi
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024


```
while [ condition ]
do
   commande1
   commande2
   ...
done
```

```
until [ condition ]
do
   commande1
   commande2
   ...
done
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024


```
for  i in a b c ...
do
   commande1
   commande2
   ...
done
```

```
for  i in *
do
   commande1
   commande2
   ...
done
```

```
for (( i=1; i<=N; i++ ))
do
   commande1
   commande2
   ...
done
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

Bash
case და select

```
case i in
expression1)
    commande1 ;;
expression2)
    commande2 ;;
...
esac
```

```
select  i in a b c d ...
do
     commande1
     commande2
     ...
done
```

```
[0-9]=[[:digit:]]; [a-z]=[[:lower:]];  [A-Z]=[[:upper:]]; PS3
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

Bash
ფუნქციები, შიდა ცვლადები

`$0, $1, ... ${10}, ${11}, $#, $*, $$, $?, $SECONDS, $RANDOM, $REPLY, read, sleep, break [N], continue [N], exit, shift, beep, getopts, $OPTARGS, debug : bash -x script`

შემთხვევითი რიცხვი ინტერვალიდან: `shuf -i n-m -n1` მაგ :`(shuf -i 100-500 -n 1)`

```
function nom {
    commande1
    commande2
    ...
}
```

```
nom () {
    commande1
    commande2
    ...
}
```

```
while getopts "c:sd" option
do
case $option in
   s) echo "-s" ;;
   d) echo "-d" ;;
   c) echo "-c $OPTARG" ;;
   *) echo "Not recognized argument"; exit -1 ;;
esac
done
```

`alias alias_name='commands'. unalias.`
პროირიტეტი : **alias ⇨ fonctions ⇨ commandes internes ⇨ commandes externes.**

Cursor Movement Capabilities :

`tput cup Y X` : Move cursor to screen location X,Y (top left is 0,0)

`tput sc` : Save the cursor position

`tput rc` : Restore the cursor position

`tput lines` : Output the number of lines of the terminal

`tput cols` : Output the number of columns of the terminal

`tput cub N` : Move N characters left

`tput cuf N` : Move N characters right

`tput cuu N`: up N lines

`tput cud N` : down N lines

tput Colour Capabilities :

`tput setab [1-7]` : Set a background colour using ANSI escape

`tput setb [1-7]` : Set a background colour

`tput setaf [1-7]` : Set a foreground colour using ANSI escape

`tput setf [1-7]` : Set a foreground colour

Text Mode Capabilities:

`tput bold` : Set bold mode

`tput dim` : Turn on half-bright mode

`tput smul` : Begin underline mode

`tput rmul` : Exit underline mode

`tput rev` : Turn on reverse mode

`tput smso` : Enter standout mode (bold on rxvt)

`tput rmso` : Exit standout mode

`tput sgr0` : Turn off all attributes

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ANSI / VT100 ტერმინალების კოდები

Ressource : http://misc.flogisoft.com/bash/tip_colors_and_formatting#colors2

ANSI / VT100 ტერმინალებსა და ტერმინალის ემულატორებს შავ თეთრის გარდა შეუძლიათ სხვა ფერებთან მუშაობაც. აქ გამოიყენება აკონტროლო სიმბოლო - <ESC>={\e,\033,\x1B}. სინტაქსური ჩანაწერი ასეთია :

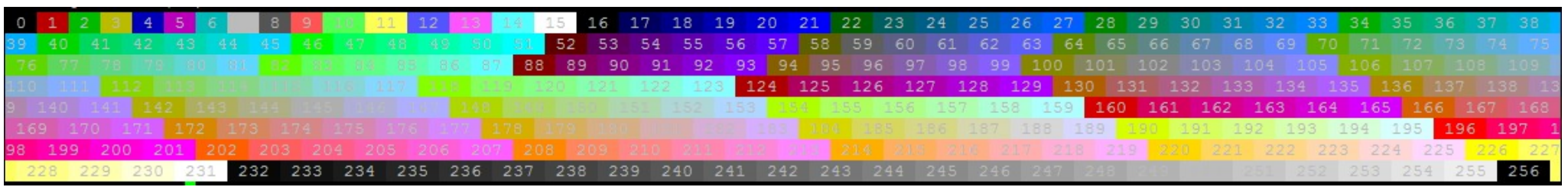
```
echo -e "\e[fg_code;bg_code;format_codem TEXT \e[0m" ან echo -e "\033[format_code;fg_code;bg_codem TEXT \033[0m" ან
echo -e "\x1B[format_code;fg_code;bg_codem TEXT \x1B[0m"
```

fg_code={30,31,32,33,34,35,36,37,39(ნაგულისხმევი ფერები),90,91,92,93,94,95,96,97(უფრო ნათელი ფერები)}
bg_code={40,41,42,43,44,45,46,47,49(ნაგულისხმევი ფერები),100,101,102,103,104,105,106,107(უფრო ნათელი ფერები)}

format_code={1-**Bold**, 2-Dim, 4-Underlinded, 5-Blink, 7-Reverse, 8-Hidden (usefull for passwords), 0-Reset all attributes, 21-Reset bold, 22-Reset dim, 24-Reset underlined, 25-Reset blink, 27-Reset reverse, 28-Reset hidden}

256 ფერის გამოყენება (ტექსტის ფერი):echo -e "\e[38;5;256_color_codem TEXT \e[0m"
256 ფერის გამოყენება (ფონის ფერი) :echo -e "\e[48;5;256_color_codem TEXT \e[0m"

256_color_code={0..256} (gnome-terminal, Konsole ... აქვთ ამის მხარდაჭერა)



LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

კლავიშების ცნობილი კომბინაციები

კურსორის მოძრობა:

Ctrl + a ხაზის დასაწყისში გადასვლა
Ctrl + e ხაზის ბოლოში გადასვლა
Ctrl + p წინა ბრძანება (ისარი ზევით)
Ctrl + n შემდეგი ბრძანება (ისარი ქვევით)
Alt + b ერთი სიტყვით უკან (მარცხნივ)
Alt + f ერთი სიტყვით წინ (მარჯვნივ)
Ctrl + f წინ ერთი სიმბოლოთი
Ctrl + b უკან ერთი სიმბოლოთი
Ctrl + xx ხაზის დასაწყისსა და კურსორის მიმდინარე პოზიციის გადართვა

პროცესების კონტროლი:

Ctrl + C გაშვებულის შეწყვეტა/მოკვლა (SIGINT)
Ctrl + l ეკრანის გასუფთავება
Ctrl + s ეკრანზე გამოტანის შეჩერება
Ctrl + q ეკრანზე გამოტანის გაგრძელება (თუ შეჩერებულია)
Ctrl + D EOF მარკერის გაგზავნა. ხურავს შელს (EXIT)
Ctrl + Z SIGTSTP სიგნალის გაგზავნა, გაშერება

რედაქტირება:

Ctrl + L ეკრანის გასუფთავება ;
Alt + d კურსორის შემდეგ სიტყვის წაშლა
Ctrl + d სიმბოლოს წაშლა კურსორის პოზიციაზე
Ctrl + h სიმბოლოს წაშლა კურსორის წინ (Backspace)
Ctrl + w კურსორამდე სიტყვის ამოჭრა
Ctrl + k კურსორის შემდეგ ხაზის ამოჭრა
Ctrl + u კურსორამდე ხაზის ამოჭრა
Alt + t მიმდინარე და წინა სიტყვების გადანაცვლება
Ctrl + t კურსორამდე ორი სიმბოლოს გადანაცვლება (typo)
Esc + t კურსორამდე ორი სიტყვის გადანაცვლება
Ctrl + y ბოლო ამოჭრილის კოპირება (yank)
Alt + u კურსორიდან სიტყვის ბოლომდე ასოების გადიდება
Alt + l კურსორიდან სიტყვის ბოლომდე ასოების დაპატარავება
Alt + c მიმდინარე სიმბოლოს გადიდება და სიტყვის ბოლოში გადასვლა
Ctrl + _ ბოლოს გაუქმება (Ctrl-x Ctrl-u იგივეა)
TAB ფაილების სახელების შევსება

ისტორია:

Ctrl + r ბოლო ბრძანების ძებნა და გამოძახება
Ctrl + p წინა ბრძანება ; Ctrl + n შემდეგი ბრძანება
!! ბოლო ბრძანების გამოძახება
!abc ბოლო ბრძანების გამოძახება, რომელიც იწყება abc-თი
!abc:p ბოლო ბრძანების ნახვა, რომელიც იწყება abc-თი
!N მე-N ბრძანების გაშვება
!$/!^ წინა ბრძანების ბოლო/პირველი არგუმენტი
ALT + . წინა ბრძანების ბოლო არგუმენტი
!* წინა ბრძანების ყველა არგუმენტი
^abc^xyz წინა ბრძანების გაშვება, სადაც abc შეცვლილია xyz-თი

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

ტერმინალის მულტიფლექსერები keybinding

Ressource : https://www.tecmint.com/screen-command-examples-to-manage-linux-terminals/
Ressource : http://lukaszwrobel.pl/blog/tmux-tutorial-split-terminal-windows-easily

Tmux
Cmd ...
Ctrl-B " - პანელის გაყოფა ვერტიკალურად
Ctrl-B % - პანელის გაყოფა ჰორიზონტალურად
Ctrl-B < ←,↑,→,↓ > - კურსორის მოძრაობა
Ctrl-B d – სესიიდან გამოსვლა

tmux ls - tmux სესიების სია
tmux attach -t N - მე-N სესიაზე მიერთება
tmux kill-session -t N – მე-N სესიის დამთავრება

screen
Cmd ...
Ctrl-A d – screen-დან გამოსვლა
Ctrl-A ? - screen-ის პარამეტრები
screen -r [pid.tty.host] - მიერთება
screen -ls – სიის ნახვა

screen-ზე პაროლის დადება

mkpasswd mot-de-passe => wM165awOh0O6M

$HOME/.screenrc-ში
Password wM165awOh0O6M

როგორ შევქმნათ ლინუქსის მაკროსი და კლავიშების კომბინაციის ფუნქცია ?

მაკროსი – შესაძლებელია კლავიშების კომბინაციია განსაზღვრა, რომელიც, როდესაც ის შესრულება, შეავსებს ასო-ნიშნების განსაზღვრულ წყობით კურსორის პოზიციას. ამისთვის, შემდეგი ფორმატის ხაზი უნდა დავამატოთ .inputrc კონფიგურაციის ფაილში:
<key combination>: "<string of characters>" ; მაგ : "\C-g": " > ~/debug_output.txt"

ფუნქცია – შესაძლებელია კლავიშების კომბინაციამ გამოიძახოს გარკვეული ფუნქცია. წინასწარ განსაზღვრული ფუნქციის მიბმა :
bind -x '<key combination>: <function-name>' ; მაგ : bind -x '"\C-j": . ~/myscript.sh'
წინასწარ განსაზღვრული ფუნქციების სია -
http://www.gnu.org/software/bash/manual/html_node/Bindable-Readline-Commands.html#Bindable-Readline-Commands

bind -p – არსებული კლავიშების კომბინაციები.

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

Here documents
here strings

Here Documents : ამ ტიპის გადამისამართება შელს ეუბნება წაიკითხოს მიმდინარე წყაროდან მანამ, სანამ ხაზში მხოლოდ ეს გამოსახულება არ დახვდება.

`<<word` (`<<-word`: ტაბულაციებით დაწყებული ხაზებში ამ ტაბულაციების უგულებელყოფა მოხდება)

```
tr a-z A-Z << END_TEXT
one two three
four five six
END_TEXT
```

მისი შედეგია :

```
ONE TWO THREE
FOUR FIVE SIX
```

```
tr a-z A-Z <<- END_TEXT
 one two three
 four five six
END_TEXT
```

მისი შედეგია :

```
ONE TWO THREE
FOUR FIVE SIX
```

```
cat << EOF
\$ Home dir "$PWD" `pwd`
EOF
```

მისი შედეგია :

```
$ Home dir "/home/user" /home/user
```

Here strings : არსებობს Bash-ში, ksh-სა და zsh-ში. `<<<` `word`-ით შესასვლელი განისაზღვრება მოცემული `word`-ით.

```
tr a-z A-Z <<< 'one two three'
```

მისი შედეგია :

```
ONE TWO THREE
```

```
FOO='one two three'
tr a-z A-Z <<< $FOO
```

მისი შედეგია :

```
ONE TWO THREE
```

```
bc <<< 2^10
```

მისი შედეგია :

```
1024
```

login shell: A login shell-ით მომხმარებელი შედის სისტემაში. მისი წყარო ფაილებია:

`/etc/profile` და `~/.profile` Bourne shell-თან თავსებად შელებისთვის (და `/etc/profile.d/*`)

`~/.bash_profile` bash-სთვის.

`/etc/zprofile` და `~/.zprofile` zsh-სთვის. `/etc/csh.login` და `~/.login` csh-სთვის.

non-login shell: მისი წყარო ფაილებია:

`/etc/bashrc` და `/.bashrc` bash-სთვის.

interactive shell: შელი (login ან non-login), სადაც შესაძლებელია ბრძანებების შეწყვეტა ინტერაქტიულად. მაგალითად gnome terminal (non-login) ან virtual terminal (login). ასეთ შელში უნდა არსებობდეს `$PS1`. მისი წყარო ფაილებია:

`/etc/profile` და `~/.profile`

`/etc/bashrc` ან `/etc/bash.bashrc` bash-სთვის.

non-interactive shell: ავტომატურად გაშვებული პროცესის მიერ შექმნილი ქვეშელია, სადაც არც შესასვლელი და არც გამოსასვლელი ჩანს. სკრიფტის გაშვებისას უკვე არა ინტერაქტიული შელი გვაქვს, მაგრამ სკრიპფს შეუძლია იმტერაქტივის ემულირება მომხმარებლის მიერ კლავიატურიდან აკრეფილი მონაცემის შეტანის დროს შესასვლელზე. მისი წყარო ფაილებია:

`/etc/bashrc` ან `/etc/bash.bashrc` bash-სთვის (უმეტეს შემთხვევაში სკრიფტის დასაწყისში არის ხოლმე ასეთი ჩანაწერი: `[ -z $PS1 ] && return`. ეს ნიშნავს, რომ არაინტერაქტიული შელის დროს გამოვიდეს (არაფერი შესრულდეს))

ზოგიერთ შემთხვევაში ის კითხულობს ფაილს, რომელიც მოცემულია `$ENV` ცვლადში.

MTA (Mail transfer Agent) ელ.ფოსტის გაგზავნა შელიდან

sendmail; postfix; qmail; exim

```
mail user@domain.org (ინტერაქტიული რეჟიმი); echo "body text" | mail -s "subject" user@domain.org

cat body_message_file | mutt -s "subject" user@domain.org -a attachment_file
mutt -s "subject" user@domain.org -a attachment_file <[body_message_file, /dev/null]

uuencode input-attachment.txt output-attachment.txt | mail -s "subject" user@domain.org
uuencode -m input-attachment.txt output-attachment.txt | mail -s "subject" user@domain.org (-m: base64 attachments)

cat body-message.txt; uuencode input-attachment.txt output-attachment.txt | mail -s "subject"  user@domain.org
```

ssmtp (gmail-ი ნიმუში); sSMTP არის მარტივი MTA ელ. ფოსტების კომპიუტერიდან mail hub-ისკენ (SMTP სერვერი) გადასაგზავნად. sSMTP-თი ვერ მივიღებთ ფოსტას.

/etc/ssmtp/ssmtp.conf :

```
        mailhub=smtp.gmail.com:587
        AuthUser=user_gmail
        AuthPass=password
        FromLineOverride=YES
        UseSTARTTLS=YES
```

mail transport სისტემები თავდაპირველად აგზავნიდა სიმბოლოებს 7 ბიტიანი კოდირებით -- როგორიცაა ASCII. ანუ, მას შეეძლო ტექსტური ფაილების გაგზავნა და არა სხვა ფაილების ("binary" text), მაგალითად სურათის (8 ბიტიანი კოდირებით). ამ შეზღუდვის მოსახსნელად გამოიყენება პროგრამა - uuencode ("UNIX to UNIX encoding"), რომელსაც გადაყავს ელ.ფოსტა binary ფორმატიდანt text ფორმატში. პირიქით ოპერაცია კი – uudecode პროგრამა აკეთებს.

Ressource : https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/

`/proc/net` ; ქსელის ინტერფეისები (eth0, eth1 ..., lo)

systemd/udev v197-დან დაწყებული ახალი დასახელებები :

- დასახელება, რომელიც შეიცავს Firmware/BIOS-სგან მოწოდებულ ინტეგრირებული მოწყობილობების ინდექსირებულ ნომბრებს (მაგ: eno1)
- დასახელება, რომელიც შეიცავს Firmware/BIOS-სგან მოწოდებულ PCI Express hotplug slot-ის ინდექსირებული ნომბრებს (მაგ: ens1)
- დასახელება, რომელიც შეიცავს მოწყობილობის ფიზიკურ/გეოგრაფიულ ადგილმდებარეობას (მაგ: enp2s0)
- დასახელება, რომელიც შეიცავს MAC მისამართს (მაგ: enx78e7d1ea46da)
- ბირთვის კლასიკური ფორმატი (მაგ: eth0)
- ქსელის ინტერფეისის სახელი იქნება 1) თუ ეს ინფორმაცია წვდომადია, თუ არაა წვდომადი , მაშინ 2), თუ ესეც არ არის, მაშინ 3) შემდეგ 5) და თუ არც ეს არსებობს, მაშინ 4)

Ressource : https://www.cyberciti.biz/faq/linux-ip-command-examples-usage-syntax/

```
ifconfig <=> ip addr show/list; ip a s; ip a l; ip a; ip -4/-6 a (IPv4/6)

ifconfig eth0 IP [netmask masque] <=> ip addr add/del IP[/masque] dev eth0

ifconfig eth0 up/down <=> ip link set dev eth0 up/down

route [-n] <=> ip route show/list ip r s; ip r l; ip r; ip -4/-6 r (IPv4/6)

route add/del -net IP eth0 <=> ip route add/del IP/masque dev eth0

route add/del -host IP gw IP_gw <=> ip route add/del IP via IP_gw

route add delfault gw IP <=> ip route add default via IP

arp <=> ip neigh show; ip neigh list; ip n s; ip n l; ip n
```

RHEL/CentOS/Fedora:

`/etc/sysconfig/network-scripts/ifcfg-eth0`

```
DEVICE="eth0"
BOOTPROTO=static
ONBOOT=yes
TYPE="Ethernet"
IPADDR=192.168.50.2
NAME="System eth0"
HWADDR=00:0C:29:28:FD:4C
GATEWAY=192.168.50.1
```

Persistant route:

```
vi /etc/sysconfig/network-scripts/route-eth0

10.10.20.0/24 via 192.168.50.100 dev eth0
```

Ubuntu/Debian/Linux Mint:

`/etc/network/interfaces`

```
auto eth0
iface eth0 inet static
address 192.168.50.2
netmask 255.255.255.0
gateway 192.168.50.1
```

```
/etc/init.d/networking restart
```

Persistant route: (`/etc/network/interfaces`)

```
up ip route add 10.10.20.0/24 via 192.168.50.100
dev eth0

up route add -net 10.10.20.0 netmask 255.255.255.0
gw 192.168.50.100
```

DHCP (Dynamic Host Configuration Protocol) პროცესი :

DHCP კლიენტი ქსელში აგზავნის broadcast შეტყობინებას რომ აღმოაჩინოს DHCP სერვერი.

ერთი ან მეტი DHCP სერვერი უპასუხებს საკუთარი broadcast შეტყობინებით და შესთავაზებს IP მისამართს.

კლიენტი აირჩევს ერთ-ერთ სერვერს დასტურის გაგზავნით .

არჩეული სერვერი ჩაინიშნავს ამ კავშირს ჟურნალში. სხვა სერვერები არაფერს აკეთებენ, რადგან მათ უარი მიიღეს.

/etc/dhcpd.conf

```
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.200 192.168.1.204;
  default-lease-time 600;
  option subnet-mask 255.255.255.0; option broadcast-address 192.168.1.255;
  option routers 192.168.1.1;  option domain-name-servers 192.168.1.25;
}
```

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

პორტები, პროტოკოლები ...

`/etc/services` (1-1023 root ; 1024-49151 – დარეზერვირებული პორტები (კომპანიის/პირების პროგრამებისთვის), 49152-65535 რეგულარული მოხმარება)

`/etc/protocols`

`/etc/hosts (IP  hostname  hostname.domainname)`

`/etc/nsswitch.conf`

`/etc/host.conf - (order hosts,bind : რეზოლვერის შემოწმება – ჯერ /etc/hosts, შემდეგ DNS; multi on : მრავალი IP მისამართების შესაძლებლობა ერთ ჰოსტზე)`

`/etc/resolv.conf - (nameserver IP; search/domain domainname)`

`/etc/networks - (loopback 127.0.0.0; mylan 192.168.1.0)`

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

სასარგებლო პრძანებები
nmap, netstat

Ressource : https://www.tecmint.com/nmap-command-examples/
Ressource : https://www.tecmint.com/20-netstat-commands-for-linux-network-management/

NMAP (**Network Mapper**) გამოიყენება ქსელების გამოსაკვლევად. ის ასკანირებს პორტებს დისტანციურ აღმოსაჩენად.

`nmap hostnames/IPs` – ასკანირებს მოცემულ მანქანას/IP მისამართს; (`-v`: დეტალური ინფორმაცია)
`nmap 192.168.0.*` - ასკანირებს მთლიან ქსელს; `nmap 192.168.0.* --exclude 192.168.0.5` – ასკანირებს გარდა...
`nmap 192.168.0.101,102,103` – ასკანირებს რამდენიმე IP-ს; `nmap 192.168.0.101-110` – ასკანირებს ინტერვალს
`nmap -iL liste.txt` – ასკანირებს ჰოსტებს ფაილიდან
`nmap -F IP` – სწრაფი სკანირება (`-T4`: fast execution) ; `nmap -sU IP` – ასკანირებს UDP პორტებს
`nmap -p 80 IP` – ასკანირებს მოცემულ პორტს მოცემულ IP-ზე; `nmap -p 80,443`; `nmap -p 80-90 IP`
`nmap --iflist` – გამოიტანს ჰოსტის ინტერფეისისა და მარშუტიზაციის ინფორმაციას
`nmap -sS IP` – აწარმოებს დაფარულ სკანირებას (ე.წ. stealthy Scan) (ეს ტექნიკას ხშირად უწოდებენ ნახევრად ღია სკანირებას, რადგან ამ დროს არ ხსნით სრულ TCP კავშირს)
`nmap -sP 192.168.0.0/24` – რომელი ჰოსტის ჩართულია (იგივეა, რაც: `nmap -sP 192.168.0.*`)

Netstat (Network Statistics) მონიტორინგს უწევს ქსელურ კავშირებს, გვანახებს მარშუტიზაციის ცხრილებს, ინტერფეისების სტატისიკებს და ა.შ.

`netstat -a` - გვანახებს ყველა ღია პორტს; `netstat -at` - მხოლოდ ღია TCP პორტებს; `netstat -au` - UDP პორტებს
`netstat -l` - გვანახებს ყველა აქტიურ კავშირს; `-p` - ე.წ. program mode. გამოაქვს პროცესის ID (PID) და სახელი
`netstat -s` - სტატისტიკა პროტოკოლების მიხედვით;
`netstat -r` - მარშუტიზაციის ცხრილი ; `netstat -rn` - მარშუტიზაციის ცხრილი (numeric mode-ში)

Ressource : http://www.thegeekstuff.com/2009/11/ping-tutorial-13-effective-ping-command-examples/ Ressource : https://www.tecmint.com/learn-how-to-use-fuser-command-with-examples-in-linux/ ; Ressource : http://www.thegeekstuff.com/2012/02/dig-command-examples ; Ressource : https://hakin9.org/syn-flood-attacks-how-to-protect-article/ ; Ressource : http://searchsecurity.techtarget.com/definition/SYN-flooding

`PING` - (Packet INternet Groper). Ping იყენებს ICMP პროტოკოლს (Internet Control Message Protocol)

- `ping IP/hostname` – შეამოწმოს შართულია თუ არა ჰოსტი
- `ping 0` - Ping localhost. (იგივეა, რაც: `ping localhost; ping 127.0.0.1`)
- `ping -i 5 IP/hostname` – Ping-ის ინტერვალის დროის გაზრდა. ჩემდეგი პაკეტი გაუშვას 5 წამის შემდეგ
- `ping -c 4 IP/hostname` - N პაკეტის გაშვება
- `ping -f IP/hostname` – ადმინისტრატორს შეუძლია ათასობით პაკეტის გაგზავნა -f ოფციით

`traceroute IP/hostname` – ჰოსტამდე გზის გაგება. In Linux, traceroute by default sends a sequence of User Datagram Protocol (UDP) packets addressed to a destination host; ICMP Echo Request or TCP SYN packets can also be used. In Windows, traceroute sends ICMP echo requests instead of UDP packets. The time-to-live (TTL) value, also known as hop limit, is used in determining the intermediate routers being traversed towards the destination. Routers decrement TTL values of packets by one when routing and discard packets whose TTL value has reached zero, returning the ICMP error message ICMP Time Exceeded. Common default values for the initial TTL are 128 (Windows OS) and 64 (Unix-based OS). `traceroute -I IP/hostname` - Use ICMP ECHO for probes.

`mtr IP/hostname; telnet IP/hostname port; host [-v] IP/hostname [another dns_server]; whois IP/hostname;`

`nslookup [-query=mx/ns/soa/any] IP/host [another dns_server]; nslookup >set type=[mx/ns/soa/any] IP/host [another dns_server]`

`dig domainname [@from_another dns_server]; dig -x IP` (PTR record trace); `dig MX tsu.ge`

`FPING` – ეს პროგრამაც ამოწმებს, თუ რომელი ჰოსტია ჩართული. `Ping`-სგან განსხვავებით მას შეიძლება ბევრი არგუმენტი გადავცეთ.

- `fping -a -g 192.168.2.0/24` – ქსელის შემოწმება
- `fping -s -g 192.168.2.1 192.168.2.11 -r 1` – ინტერვალის შემოწმება

ACK (Acknowledgement), **A three-way handshake**, **SYN flood** (half open attack). SYN => SYN/ACK => ACK ; DoS (denial-of-service)

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

lsof, fuser

Ressource : http://www.thegeekstuff.com/2012/08/lsof-command-examples
Ressource : https://www.tecmint.com/learn-how-to-use-fuser-command-with-examples-in-linux/

`LSOF` - **L**ist **O**pen **F**iles. გამოაქვს პროცესების მიერ გახსნილი ფაილები.

FD – Represents the file descriptor. Some of the values of FDs are: `cwd` – Current Working Directory; `txt` – Text file; `mem` – Memory mapped file; `mmap` – Memory mapped device (For certain devices, such as frame buffers, application programs having direct access to device memory is more efficient than byte-stream I/O); `NUMBER` – Represent the actual file descriptor. The character after the number i.e '1u', represents the mode in which the file is opened. `r` for read, `w` for write, `u` for read and write

TYPE – Specifies the type of the file. Some of the values of TYPEs are: `REG` – Regular File; `DIR` – Directory; `FIFO` – First In First Out; `CHR` – Character special file

`lsof` – ყველა აქტიური პროცესის მიერ გახსნილი ფაილების სია
`lsof +D /var/log/` - გახსენილი ფაილების ამ დირექტორიიდან
`lsof -c ssh -c init` – გახსნილი ფაილები მოცემული პროცესების მიერ (პროცესის დასახელება იწყება...)
`lsof -u username` – გახსნილი ფაილები მომხმარებლის მიერ; `lsof -u ^username` – username-ის გარდა
`lsof -c bash -u user` – გახსნილი ფაილები `user`-ის მიერ **ან** `bash` პროცესის მიერ;
`lsof -c bash -a -u user` - გახსნილი ფაილები `user`-ის მიერ **და** `bash` პროცესის მიერ;
`lsof -p PID` – პროცესის მიერ გახსნილი ფაილები
`kill -9 `lsof -t -u user`` - მომხმარებლის მიერ გაშვებული ყველა პროცესის მოკვლა
`lsof -i` – ყველა ქსელური კავშირი; `lsof -i4` - IPv4 კავშირი; `lsof -i6` - IPv6 კავშირი
`lsof -i :22` – ყველა პროცესი, რომელიც უსმენს ამ პორტს; `lsof -i @IP` – ყველა კავშირი, რომელიც იყენებს ამ IP-ს
`lsof -i tcp; lsof -i udp` – ყველა TCP ან UDP კავშირი; `lsof -i -P` – პორტის ნომრები სახელების ნაცვლად

`FUSER` – გამოაქვს ინფორმაცია იმ მომხმარებლისა და პროცესის შესახებ, ვინც იყენებს მოცემულ ფაილს. წვდომის ტიპები : `c` – მიმდინარე დირექტორია; `e` - გაშვება; `f` – ფაილის გახსნა; `F` – ფაილის გახსნა ჩასაწერად; `r` – root დირექტორია; `m` - mmaped ფაილი ან გაზიარებული ბიბლიოთეკა (shared library)
`fuser .` - ვინ იყენებს დირექტორიას?; `fuser -v .` - დეტალური ინფორმაცია
`fuser -v -n tcp 22` – რომელი პროცესები იყენებენ TCP/UDP სოკეტს
`fuser -v -k file` – იმ პროცესის მოკვლა, რომელსაც იყენებს მოცემული ფიალი

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

სასარგებლო პრძანებები
netcat

Ressource : http://www.thegeekstuff.com/2012/04/nc-command-examples/

NC - (Netcat) გამოიყენება ქსელის შესასწავლად და შესამოწმებლად.
nc (-l: უსმენს შემომავალ კავშირს. -v: დეტალური ინფო; -4: IPv4 მხოლოდ; -6: IPv6 მხოლოდ; -u: UDP კავშირი TCP-ის ნაცვლად; -z: მხოლოდ პორტის დასკანირება (-v ოფციასთან ერთად); -n: ნომერი სახელის ნაცვლად)

Netcat სერვერ-კლიენტის არქიტექტურაში:
სერვერი: `nc -l -p 8888`
კლიენტი: `nc server_IP 8888`

Netcat ფაილების გადასაცემად:კლიენტი
სერვერი: `nc -l -p 8888 >file`
კლიენტი: `cat file_to_copy | nc server_IP 8888`

Port-ის ინტერვალის დასკანირება:
კლიენტი: `nc -zv server_IP 8888 10-200`

Chat Netcat-ით:
სერვერი: `nc -lp 8888`
DATA1
DATA2_reply
კლიენტი: `nc server_IP 8888`
DATA1_reply
DATA2

Netcat როგორც მარტივი Web სერვერი

სერვერი: `emacs index.html`

```
<html>
        <head>
                <title>სათაური</title>
        </head>
        <body>
                <h1>Level 1 header</h1>
                <h2>Subheading</h2>
                <p>ტექსტი</p>
        </body>
</html>
```

1) `nc -l -p 8888 < index.html`
2) `echo -e 'HTTP/1.1 200 OK\n\n%s' "$(cat index.html)" | nc -l -p 8888 -q 1`
3) `while true; do echo -e "HTTP/1.1 200 OK\n\n $(free; df -h; ps aux)" | nc -l -p 8888 -q 1; done`

კლიენტი:
1) `links server_IP 8888` (http://server_IP:8888)
2) mozilla 3) http://server_IP:8888

LINUX სისტემები

არჩილ ელიზბარაშვილი, 2024

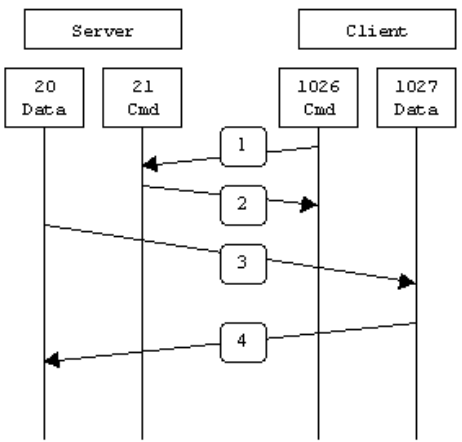
FTP პროტოკოლი

Ressource : http://www.slacksite.com/other/ftp.html
Ressource : https://www.cs.colostate.edu/helpdocs/ftp.html
Ressource : https://danielmiessler.com/study/url-uri/#gs.cK_if2g

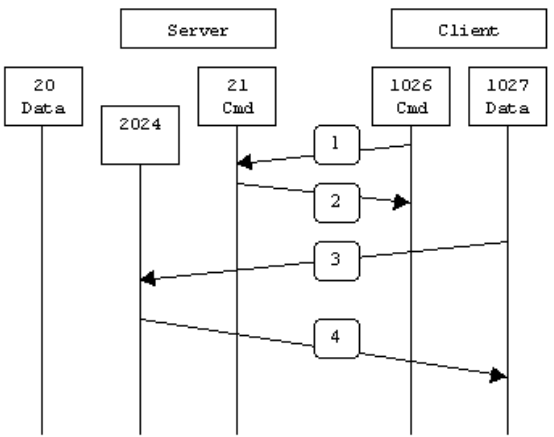
FTP არის მხოლოდ TCP-ზე დაფუძნებული სერვისი. ის იყენებს ორ პორტს - '**data**' და '**command**', 20 და 21 პორტებს.

`ftp hostname;` ftp commands: `?` (help), `ls` (list files), `put` (copy one file to), `get` (copy one file from), `ascii` (to set the mode of file transfer to ASCII (this is the default and transmits seven bits per character)), `binary` (to set the mode of file transfer to binary (the binary mode transmits all eight bits per byte and thus provides less chance of a transmission error and must be used to transmit files other than ASCII files)) . . .

Active FTP



Passive FTP



Protocol://[User[:Password]@]Host[:Port][/Path]

URI (Uniform Resource Identifier)
`Ex: ftp://ftp.ifg.tsu.ge/Cours/Linux/grep.pdf`
(Also a URL because of the protocol)

`Ex: Tel:+995322300196`

URL (Uniform Resource Locator)
`Ex: ftp://ftp.ifg.tsu.ge/Cours/Linux/grep.pdf`

URN (Uniform Resource Name)
`Ex: ftp.ifg.tsu.ge/Cours/Linux/grep.pdf`