1. ***What is an operating system?.***

Օպերացիոն համակարգը ծրագրային ապահովում է, որը նախատեսված է համակարգչի բոլոր սարքավորումները և ծրագրերը գործարկելու, կառավարելու և համակարգելու համար։

Այն ապահովում է օգտագործողի ինտերֆեյս (միջերես) և կառավարում է հետևյալ առաջադրանքները՝

1․ Ծրագրերի գործարկում

2, Ֆայլերի կազմակերպում

3․ Մուտքագրման/ելքի գործողություններ

ՕՀ-ն ապահովում է, որ ծրագրերը և օգտատերերը կարողանան արդյունավետորեն կիսվել համակարգի ռեսուրսներով։ Այսինքն՝ ՕՀ-ն կապում է Software-ը Hardware-ի հետ։ Առանց դրա ծրագիրը չի կարող աշխատել։

1. ***?.***

Linux-ը Unix-ի վրա հիմնված, բաց կոդով օպերացիոն համակարգ է: Ստեղծվելով Լինուս Տորվալդսի կողմից 1991 թվականին, Linux-ը դարձել է տեխնոլոգիական աշխարհի խոշոր ուժ՝ հզորացնելով ամեն ինչ՝ սկսած սմարթֆոններից և սերվերներից մինչև գերհամակարգիչներ, այդ թվում՝ ներդրված համակարգեր: Բաց կոդով օպերացիոն համակարգը այն համակարգն է, որի կոդը հասանելի է ցանկացած անձի կողմից դիտման, փոփոխման և տարածման համար։

1. ***?.***

GNU հապավումը բացվում է GNU’s Not Unix։ Սա անվճար և բաց կոդով օպերացիոն համակարգի նախագիծ է, որը սկսվել է Ռիչարդ Սթոլմանի կողմից 1983 թվականին: GNU-ի նպատակն էր ստեղծել ամբողջական Unix-անման օպերացիոն համակարգ, որը կհարգեր օգտատերերի ազատությունը:

GNU-ի մասին հիմնական կետերը.

1. Ազատ ծրագրային ապահովման շարժում.

GNU-ն "Free Software Movement"-ի հիմքն է: «free» այստեղ նշանակում է ազատություն, այլ ոչ թե անվճար: Օգտատերերն ունեն ազատություն՝

\* Գործարկել ծրագիրը ցանկացած նպատակով

\* Ուսումնասիրել և փոփոխել այն

\* Կիսվել դրանով

\* Տարածել փոփոխված տարբերակները

2. GNU-ի բաղադրիչները.

GNU նախագիծը մշակել է բազմաթիվ կարևոր գործիքներ, ինչպիսիք են՝

\* GNU Bash – հրամանային տողի shell

\* GCC – C, C++ և այլնի համար կոմպիլյատոր

\* GNU Core Utilities (coreutils) – հրամանային տողի հիմնական գործիքներ, ինչպիսիք են՝ «ls», «cp», «mv», «cat»

\* GDB – debugger (վրիպազերծիչ)

\* Make – կառուցման ավտոմատացման գործիք

3. Միջուկի (Kernel) խնդիրը.

GNU-ն ցանկանում էր ստեղծել լիարժեք օպերացիոն համակարգ, ներառյալ Kernel (անվանումը GNU Hurd), բայց այն երբեք ավարտին չի հասցվել կամ լայնորեն չի ընդունվել։

4. GNU/Linux:

Երբ Լինուս Տորվալդսը 1991 թվականին թողարկեց Linux Kernel-ը, այն համակցվեց GNU գործիքների հետ՝ լիարժեք աշխատող օպերացիոն համակարգ ստեղծելու համար։ Ահա թե ինչու շատերը օպերացիոն համակարգը անվանում են՝ GNU/Linux։

Օրինակ: Եթե օգտագործում ենք Ubuntu, Debian կամ Fedora, ապա օգտագործում եք GNU/Linux համակարգ՝ GNU գործիքներ + Linux Kernel։

1. ***?.***

Unix ճարտարապետության չորս հիմնական բաղադրիչներն են՝

1. Hardware

2. Kernel

3. Shell

4. Applications

1. Hardware շերտը ներառում է համակարգչի բոլոր ֆիզիկական բաղադրիչները՝ օպերատիվ հիշողություն (RAM), կոշտ սկավառակ(HSS), պրոցեսոր(CPU) և մուտքային/ելքային սարքեր։

2. Kernel. Kernel-ը Linux-ի վրա հիմնված օպերացիոն համակարգի միջուկն է։ Այն վիրտուալացնում է համակարգչի ընդհանուր սարքային ռեսուրսները՝ յուրաքանչյուր գործընթացին տրամադրելով իր վիրտուալ ռեսուրսները։ Սա գործընթացը դարձնում է այնպիսի տպավորություն, կարծես այն մեքենայի վրա աշխատող միակ գործընթացն է։ Kernel-ը նաև պատասխանատու է տարբեր գործընթացների միջև կոնֆլիկտները կանխելու և մեղմելու համար։ Միջուկի տարբեր տեսակներն են՝

•Monolithic Kernel

•Hybrid kernels

•Exo kernels

•Micro kernels

3․ Shell. Shell-ը Linux օպերացիոն համակարգի օգտագործողի ինտերֆեյսն է: Այն թույլ է տալիս օգտատերերին փոխազդել համակարգի հետ՝ մուտքագրելով հրամաններ, որոնք Shell-ը մեկնաբանում և կատարում է: Shell-ը ծառայում է որպես կամուրջ օգտատիրոջ և Kernel-ի միջև՝ փոխանցելով օգտատիրոջ հարցումները Kernel-ին՝ մշակման համար: Այն օգտատերերին հնարավորություն է տալիս կատարել տարբեր առաջադրանքներ, ինչպիսիք են ծրագրերի գործարկումը, ֆայլերի կառավարումը և համակարգի կարգավորումը:

4. Commands (Հրամաններ) and Utilities (Ծառայություններ) - Կան տարբեր Command-ներ և Utilities-ներ, որոնք կարող եք օգտագործել ձեր ամենօրյա գործունեության մեջ: cp, mv, cat, grep և այլն: Կան ավելի քան 250 ստանդարտ հրամաններ, գումարած բազմաթիվ այլ հրամաններ, որոնք տրամադրվում են երրորդ կողմի ծրագրային ապահովման միջոցով:

Files and Directories - Unix-ի բոլոր տվյալները կազմակերպվում են ֆայլերի մեջ: Այնուհետև բոլոր ֆայլերը կազմակերպվում են դիրեկտորիաների մեջ: Այս դիրեկտորիաները հետագայում կազմակերպվում են ծառի նման կառուցվածքի մեջ, որը կոչվում է ֆայլային համակարգ:

1. ***?.***

1. Free and Open Source

2. Security

3. Stability and Reliability

4. Performance and Efficiency

5. Used in Servers, Cloud, and Embedded Systems

6. Strong Community Support

7. Support for Programming

1. ***?.***

1. Software Compatibility

Որոշ հայտնի առևտրային ծրագրեր (օրինակ՝ Adobe Photoshop, MS Office, որոշ խաղեր) չեն աշխատում Linux-ի վրա։

2. Gaming Limitations

3. Learning Curve for Beginners

4. Hardware Compatibility

Some devices and drivers (especially for printers, Wi-Fi adapters, or graphics cards) may not work perfectly out of the box.

5. Less Support for Commercial Software

Ծրագրային ապահովման խոշոր ընկերություններն ավելի շատ կենտրոնանում են Windows/macOS-ի վրա:

6. Fragmentation

Linux-ի բազմաթիվ distribution-ներ աշխատում են տարբեր փաթեթային համակարգերով, գործիքներով և ֆայլային կառուցվածքներով:

Կարող է շփոթեցնել սկսնակներին և դժվարացնել ծրագրային ապահովման աջակցությունը

7. No Standardization

1. ***?.***

Fedora

Debian

Ubuntu

openSuse

CentOS

linuxmint

1. ***?.***

GPL stands for GNU General Public License. It is a free software license created by Richard Stallman for the GNU Project.

GPL-ը հապավումը բացվում է GNU General Public License: Այն անվճար ծրագրային ապահովման լիցենզիա է, որը ստեղծվել է Ռիչարդ Սթոլմանի կողմից GNU նախագծի համար:

GPLv1 (1989): Original version

GPLv2 (1991): More clear legal language; used by the Linux kernel

GPLv3 (2007): Updated to close loopholes (like "Tivoization"), better support for international law

Key Features of GPL:

1. Freedom to Use

Կարող ենք օգտագործել ցանկացած նպատակով՝ անձնական, կրթական, գործարար և այլն։

2. Freedom to Study and Modify

You can look at the source code and change it however you like.

Կարող ենք նայել source code-ը և փոխել այն, ինչպես ցանկանում եք:

3. Freedom to Share

You can distribute the original or your modified version to others.

Կարող ենք տարածել բնօրինակը կամ մեր կողմից փոփոխված տարբերակը ուրիշներին:

4. Copyleft

Սա ամենակարևոր մասն է.

Եթե ​​դուք փոփոխում և տարածում եք GPL լիցենզավորված ծրագրակազմը, ձեր տարբերակը նույնպես պետք է լինի GPL:

Դուք պետք է կիսվեք source code-ով ուրիշների հետ:

1. ***?.***

1․ GPL (General Public License)

Full copyleft license

Եթե դուք փոփոխում եք GPL լիցենզավորված կոդը կամ համատեղում այն ​​ձեր սեփական կոդի հետ, ձեր ամբողջ նախագիծը պետք է թողարկվի GPL լիցենզիայի ներքո, եթե այն տարածում եք։

Սա ապահովում է, որ բոլոր ածանցյալ աշխատանքները մնան անվճար և բաց կոդով։

2․ LGPL (Lesser General Public License)

Weaker copyleft license, ավելի թույլատրելի

Դուք կարող եք հղումներ կատարել LGPL լիցենզավորված գրադարաններին՝ առանց ձեր ամբողջ նախագիծը GPL դարձնելու։

Միայն LGPL գրադարանի փոփոխությունները պետք է լինեն բաց կոդով, եթե վերաբաշխվեն։

Ձեր սեփական կոդը (որն օգտագործում է գրադարանը) կարող է մնալ փակ կոդով կամ այլ լիցենզիայի ներքո։

Օգտագործեք GPL, եթե ցանկանում եք ստիպել, որ ձեր կոդն օգտագործող բոլոր ծրագրակազմերը մնան բաց կոդով։

Օգտագործեք LGPL, եթե ստեղծում եք գրադարան և ցանկանում եք թույլ տալ ուրիշներին օգտագործել այն ինչպես բաց, այնպես էլ փակ նախագծերում։

1. ***What are three basic file types in Linux?***

Regular Files

Directories

Special Files

1. ***?.***

Օգտագործվում է տերմինալում ls հրամանը։

1. ***?.***

\*-ը մետա-նիշ է, որն օգտագործվում է shell-ում՝ ֆայլերի և գրացուցակների անուններում զրո կամ ավելի նիշեր համապատասխանեցնելու համար։

Այն շատ օգտակար է ֆայլերը որոնելու, զտելու կամ խմբային մշակման համար։

Օրինակ։ \* => Բոլոր ֆայլերը ընթացիկ directory-ի

\*․txt => Բոլոր ֆայլերը, որոնք ավարտվում են․txt-ով

1. ***?.***

Linux-ում hidden file-ը ցանկացած ֆայլ կամ թղթապանակ է, որի անունը սկսվում է կետով (.):

Օրինակ՝ .bashrc, .git, .config, .profile։

1. ***?.***

"vim" stands for Vi IMproved.

Այն հրամանային տողի տեքստային խմբագրիչ է, որն օգտագործվում է տեքստային ֆայլեր ստեղծելու և խմբագրելու համար:

touch, Միայն ստեծում է, չես կարող գրել։

nano, Եվ ստեղծում է, և կարող ես գրել։

1. ***?.***

cat ֆայլի անունը

Օրինակ` cat file5

1. ***?.***

cp source\_file destination

1. ***?.***

mv old\_name new\_name

1. ***?.***

rm filename

1. ***?.***

Երբ դուք մուտք եք գործում Unix/Linux համակարգ, դուք սկսում եք ձեր home directory-ից։

Սա ձեր անձնական տարածքն է ֆայլերի, ծրագրերի և subdirector-ների համար։

Ցանկացած պահի ձեր home directory վերադառնալու համար. $cd~

1. ***What is root? Which files are not children of the root?***

/ is the root directory, top of the Linux file tree.

Բոլոր ֆայլերն ու թղթապանակները /-ի զավակներն են, երբ դրանք դառնում են Linux համակարգի մաս։

Even external drives appear inside / when mounted.

/bin

/etc

/home

/opt

/tmp

/usr

/var

1. ***?.***

Absolute Path

Միշտ սկսվում է /-ից։

Այն պատմում է ֆայլի կամ թղթապանակի ամբողջական ուղին, անկախ նրանից, թե որտեղ եք գտնվում։

Օրինակ: /home/alex/Documents/report.txt

Միշտ հուսալի և ճշգրիտ։

Relative Path

Հիմնված է ձեր ընթացիկ աշխատանքային directory-ի վրա:

Այն չի սկսվում /-ով:

Օրինակ: Documents/report.txt

Ավելի կարճ է, բայց աշխատում է միայն ձեր գտնվելու վայրի համեմատ:

1. ***?.***

Օգտագործեք mkdir (make directory) հրամանը.

mkdir folder\_name

Ստեղծում է myfolder անունով թղթապանակ ընթացիկ directory-ում:

mkdir /home/alex/projects/new

Ստեղծում է թղթապանակ, բացարձակ ճանապարհով:

mkdir -p parent/child/grandchild

-p-ը Linux-ին ասում է ստեղծել բոլոր մայր թղթապանակները, եթե դրանք գոյություն չունեն:

Օգտագործեք rmdir կամ rm հրամանը:

Հեռացնել դատարկ գրացուցակը.

rmdir folder\_name

Հեռացրեք directory-ն ներսում ֆայլերով.

rm -r folder\_name

-r նշանակում է ռեկուրսիվ — այն ջնջում է թղթապանակը և դրա մեջ եղած ամեն ինչ:

1. ***?.***

cd directory\_name

Օրինակ։ cd Documents

cd /home/alex/Downloads

cd (վերադարձ home directory)

1. ***?.***

. — Current Directory

Ներկայացնում է այն directory-ն, որտեղ դուք գտնվում եք ներկայումս։

Օրինակ։ cd .

.. — Parent Directory

Ներկայացնում է directory-ի մեկ մակարդակ վերևը:

Used to move or refer to the parent directory.

Օրինակ։ cd ..

1. ***?.***

The Three File Permission Attributes:

Read (r) - կարդալ

Թույլ է տալիս դիտել կամ կարդալ ֆայլի պարունակությունը։

For directories: թույլ է տալիս ցուցադրել ֆայլերը թղթապանակի ներսում:

Write (w)

Թույլ է տալիս փոփոխել ֆայլը (edit, delete, rename)։

For directories: թույլ է տալիս ավելացնել կամ հեռացնել ֆայլերը թղթապանակում:

Execute (x)

For files: թույլ է տալիս գործարկել ֆայլը որպես ծրագիր կամ սկրիպտ:

For directories: թույլ է տալիս մուտքագրել (cd-ի մեջ) directory:

1. ***?.***

Linux-ում ֆայլերի թույլտվությունները տեսնելու համար օգտագործեք ls հրամանը -l (long listing) տարբերակով`

ls -l filename

Նմանատիպ ելք է ստացվում՝

-rw-r--r-- 1 alex users 1234 Jun 18 10:00 file.txt

Առաջին նիշը:

- նշանակում է regular file

d նշանակում է directory

l նշանակում է symbolic link

Հաջորդ 9 նիշերը բաժանված են 3 խմբի՝ բաղկացած 3-ից.

1․ Owner permissions (rw-) ՝ Սեփականատիրոջ թույլտվություններ

2․ Group permissions (r--) ՝ Խմբի թույլտվություններ

3․ Others permissions (r--) ՝ Այլ թույլտվություններ

1. ***?.***

chmod [options] mode filename

1. Symbolic Method (using letters)

r = read

w = write

x = execute

u = user (owner)

g = group

o = others

a = all (user, group, others)

Օրինակ։

Ավելացնում ենք կատարման թույլտվություն սեփականատիրոջ համար՝

chmod u+x file.txt

Հեռացնում ենք գրելու թույլտվությունը խմբի և այլոց համար՝

chmod go-w file.txt

Ավելացնում է կարդալու թույլտվություն բոլորի համար՝

chmod a+r file.txt

2. Numeric (octal) Method

R - 4

W - 2

X - 1

Օրինակ: chmod 754 file.txt

User: 7 = 4+2+1 (rwx)

Group: 5 = 4+0+1 (r-x)

Others: 4 = 4+0+0 (r--)

1. ***?.***

stdin (ստանդարտ մուտք):

Մուտքային հոսք, որտեղ ծրագիրը տվյալներ է ստանում օգտատիրոջից կամ մեկ այլ ծրագրից։

Սովորաբար միացված է ձեր ստեղնաշարին։

stdout (ստանդարտ ելք):

Ելքային հոսք, որտեղ ծրագիրը ուղարկում է նորմալ ելք (արդյունքներ, հաղորդագրություններ)։

Սովորաբար ցուցադրվում է ձեր տերմինալի էկրանին։

stderr (ստանդարտ սխալ):

Ելքային հոսք, որտեղ ծրագիրը ուղարկում է սխալի հաղորդագրություններ։

Սովորաբար նաև ցուցադրվում է ձեր տերմինալի էկրանին՝ stdout-ից առանձին։

1. ***?.***

[...] Matches any one character inside brackets.

Example։ file[123].txt

matches file1.txt, file2.txt or file3.txt

[^...] or [!...] Matches any one character NOT inside brackets

Example: file[!0-9].txt

matches fileA.txt but not file1.txt

1. ***?.***

mkdir MessyDir

cd !$

touch a.jpg, b.jpg and c.jpg

touch a.txt b.txt c.txt

touch a\_v1.txt b\_v1.jpg c\_v1.bin

ls

mkdir jpeg text v1

shopt -s extglob

mv !(\*\_v1).jpg jpeg/

mv !(\*\_v1).txt text/

mv \*\_v1.\* v1/

ls -R > ~/log

Որպեսզի !(\*\_v1) կարողանանք օգտագործել, պահանջում է extended globbing, դրա համար ենք գրում "shopt -s extglob"

հրամանը։

1. ***Come up with a similar task to above one and write the solution.  
   Use the file manipulation commands, file permissions and meta-character.***

Օ Ստեղծեք DDD directory, ներսում ստեղծեք fff անունով ֆայլը։

Այնուհետև փոխեք permission-ը այնպես, որ ակարողանան կատարման թույլտվություն ստանալ user-ի համար։

mkdir DDD

cd !$

touch fff

chmod u+x fff