1. ***Untitled Question?.***

Ի՞նչ է նշանակում անվերնագիր հարց։ Անհասկանալի է հարցը․․

1. ***What is a character device?***

Character device - Device file-ի տեսակ է, որը տրամադրում է չբուֆերացված, ուղղակի, բայթ առ բայթ մուտք hardware կամ վիրտուալ սարքերին։

Միաժամանակ փոխանցվում է մեկ նիշ (բայթ) տվյալ:

Սովորաբար օգտագործվում է այնպիսի սարքերի համար, ինչպիսիք են.

Terminals (TTY)

Serial ports

Ստեղնաշարերը

Մկնիկը

1. ***What is a block device?***

Block device - Device file-ի տեսակ է, որը թույլ է տալիս բուֆերային, բլոկ առ բլոկ մուտք գործել սարքաշար, որը տվյալները պահում է ֆիքսված չափի բլոկներում: Տվյալները հասանելի են կտորներով (բլոկներով), ոչ թե բայթ առ բայթ:

Օգտագործվում է պատահական մուտքի աջակցող սարքերի համար, ինչպիսիք են՝

Կոշտ սկավառակներ (/ dev/sda)

SSD-ներ

USB կրիչներ

CD/DVD կրիչներ

Block device-ները տվյալները փոխանցում են ֆիքսված չափի բլոկներով:

Character device-ները չեն բուֆերացնում տվյալները. նրանք հոսքային հեռարձակում են այն այնպես, ինչպես այն գալիս է։

1. ***What is a link to file?***

Հղումը նման է shortcut -ի կամ alias-ի, որը մատնանշում է մեկ այլ ֆայլ։

Linux-ում կան երկու տեսակի հղումներ՝

Hard Link - Ֆայլի հղման պատճենը սկավառակի վրա:

Ցույց է տալիս անմիջապես նույն inode-ին (փաստացի ֆայլի տվյալներ):

Եթե ​​բնօրինակ ֆայլը ջնջվի, "Hard Link"-ը դեռ աշխատում է (քանի որ տվյալները դեռ կան):

Soft Link (Symbolic Link or Symlink) - Մեկ այլ ֆայլի ցուցիչ (pionter):

Աշխատում է shortcut-ի պես:

Եթե ​​բնօրինակ ֆայլը ջնջված է, "Sowt Link"-ը կոտրվում է (ցուցադրում է "No such file or directory")։

1. ***What kind of links can we create?***

Linux-ում մենք կարող ենք ստեղծել հղումների երկու հիմնական տեսակ՝

Hard Link

Soft Link

1. ***How to create a hard link?***

Linux-ում կոշտ հղում ստեղծելու համար օգտագործում ենք ln հրամանը՝ առանց որևէ option-ի՝ ln existing\_file link\_name

Օրինակ: ln report.txt report\_copy.txt

Ստեղծում է report\_copy.txt անունով կոշտ հղում, որը report.txt-ի հետ մատնանշում է նույն inode-ին։

Երկուսն էլ այժմ կիսում են ճիշտ նույն տվյալները։

Եթե մեկը խմբագրեք, փոփոխությունները կհայտնվեն մյուսում։

Եթե մեկը ջնջեք, մյուսը դեռ կաշխատի։

1. ***How to create soft link?***

Linux-ում soft հղում ստեղծելու համար օգտագործում ենք ln հրամանը -s տարբերակով՝

ln -s target\_file link\_name

Օրինակ: ln -s report.txt shortcut.txt

Ստեղծում է shortcut.txt անունով soft հղում, որը մատնանշում է report.txt-ին։

Եթե բացենք կամ գործարկենք shortcut.txt-ը, այն գործում է ինչպես report.txt-ը։

Եթե report.txt-ը ջնջվում է, shortcut.txt-ը դառնում է կոտրված հղում։

1. ***What is an inode?***

Inode-ը (index node) տվյալների կառուցվածք է, որն օգտագործվում է Linux ֆայլային համակարգերի կողմից ֆայլի մասին մետատվյալներ պահելու համար, բայց ոչ ֆայլի անունը կամ իրական տվյալները։

ls -i

Օրինակ կարող է ցուցադրել՝ 124578 report.txt

որտեղ 124578-ը report.txt ֆայլի ինոդային համարն է։

1. ***What kind of information is saved in inodes?***

Յուրաքանչյուր ֆայլ ունի inode, որը պահպանում է.

Ֆայլի տեսակը (regular file, directory, link և այլն)

Ֆայլի թույլտվությունները (rwx)

Owner (UID) (Ֆայլի սեփականատիրոջ User ID-ն)

Group (GID) (Ֆայլի խմբի Group ID-ն)

Ֆայլի չափը (բայթերով)

Timestamps՝

atime Վերջին մուտքի ժամանակը

mtime երջին փոփոխության ժամանակը

ctime Ինոդի փոփոխության ժամանակը

Հղումների քանակը (ինչքան կոշտ հղումներ են մատնանշում դրան)

Pointers to data blocks

Ինոդի համարը (Ֆայլային համակարգի ներսում եզակի ID)

1. ***What is a difference between hard and soft links?***

HARD LINK

Նույն ինոդի կրկնօրինակ հղում (նույն ֆայլը սկավառակի վրա)։ Դա նման է նույն ֆայլի մեկ այլ անվան։

Եթե դուք փոխեք մեկը, մյուսը նույնպես կփոխվի։ Ֆայլը մնում է հասանելի, նույնիսկ եթե սկզբնական անունը ջնջվի։

Չի կարող կապվել դիրեկտորիաների հետ։

Պետք է լինի նույն ֆայլային համակարգում։

Օրինակ: ln original.txt hardlink.txt

SOFT LINK

Shortcut, որը ցույց է տալիս մեկ այլ ֆայլի path-ը (ուղին):

Այն ունի իր սեփական ինոդը:

Եթե սկզբնական ֆայլը ջնջվի, soft հղումը կխափանվի:

Կարող է կապվել դիրեկտորիաների հետ:

Կարող է կապվել ֆայլային համակարգերի միջև:

Օրինակ՝ ln -s original.txt softlink.txt

1. ***Where are inodes stored?***

Ինոդները պահվում են սկավառակի վրա՝ որպես ֆայլային համակարգի կառուցվածքի մաս, այլ ոչ թե առանձին ֆայլերի կամ հիշողության մեջ։

1. ***What is a forking?***

Ֆորկինգը այն գործընթացն է, որի ընթացքում աշխատող ծրագիրը (կոչվում է process) ստեղծում է իր պատճենը։ Այս պատճենը կոչվում է դուստր պրոցես։

Դրա համար օգտագործվում է fork() համակարգային կանչը։

Օրինակ։ ls &

1. ***What does exec system call?***

exec համակարգային կանչը փոխարինում է ընթացիկ պրոցեսը նոր ծրագրով։

fork() → ստեղծում է նոր պրոցես

exec() → փոխարինում է ընթացիկ պրոցեսը մեկ այլ ծրագրով

Այսպիսով, եթե fork()-ը նման է կլոնավորման, ապա exec()-ը նման է ձևափոխման։

1. ***What is a package manager?***

"package manager"-ը ծրագրային գործիք է, որը օգնում է տեղադրել, թարմացնել, հեռացնել և կառավարել ծրագրային փաթեթները Linux համակարգում։

1. ***What kind of packages do you know?***

Linux-ում package-ը սեղմված ֆայլ է, որը պարունակում է ծրագիր և դրա տեղադրման համար անհրաժեշտ մետատվյալներ (օրինակ՝ version, dependencies, installation scripts):

Գոյություն ունեն փաթեթների ձևաչափերի երկու հիմնական տեսակներ՝ կախված Linux-ի բաշխումից.

1․ Debian-ի վրա հիմնված փաթեթներ (.deb)

Օգտագործվում են՝ Debian, Ubuntu, Linux Mint և այլն:

2․ Red Hat-ի վրա հիմնված փաթեթներ (.rpm)

Օգտագործվում են՝ Red Hat, Fedora, CentOS, RHEL:

1. ***What kind of package managers do you know?***

Linux-ում կան բազմաթիվ Package Manager-ներ՝ կախված բաշխումից, փաթեթի տեսակից․ apt (.deb ֆորմատով), dpkg (.deb ֆորմատով), yum (.rpm ֆորմատով), dnf (.rpm ֆորմատով) և այլն։

1. ***Link to the "Command translation game" with ChatGPT***

<https://chatgpt.com/share/68551fdb-e9a4-800a-9f36-0f6337a1ad89>