*Aginduen kateatzea*

Linuxen komandoak kateatzeak hainbat komando sekuentzian exekutatzeko aukera ematen du, ";" eragilea erabiliz. Adibidez, "nuevo\_directorio" izeneko direktorio bat sortu eta direktorio horretara aldatu nahi baduzu, honako komando hau erabil dezakezu:

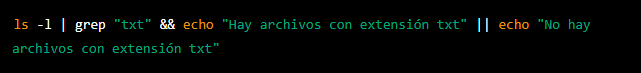


Komandoak kateatzeko beste modu bat "& &" operadorea erabiltzea da. Adibidez, "archivo.txt" izeneko fitxategi bat egungo direktorioan dagoen egiaztatu nahi baduzu eta, hala bada, haren edukia erakutsi nahi baduzu, honako komando hau erabil dezakezu:



Kasu honetan, "cat" komandoa exekutatuko da soilik "test" komandoak emaitza arrakastatsua itzultzen badu, eta horrek esan nahi du fitxategia existitzen dela.

Hona hemen adibide konplexuagoa:



Kasu honetan, egungo direktorioko fitxategiak eta karpetak zerrendatzen dira; gero, haien izenean "txt" dutenak bilatzen dira; aurkituz gero, "txt luzapena duten fitxategiak daude" inprimatzen da, "Ez dago txt luzapena duten fitxategirik" inprimatzen ez bada.

Beste adibide batzuk.

Fitxategi baten edukia erakutsi, baina hitz jakin bat duten lerroak bakarrik:



Fitxategi batean hitz bat zenbat aldiz agertzen den zenbatzea:



*Script*-ak

Linuxen script bat testu lauaren fitxategi bat da, automatikoki ordenan exekutatzen diren hainbat komando dituena. Script-ak ataza errepikakorrak automatizatzeko, ataza konplexuak urrats bakar batean egiteko eta programa pertsonalizatuak sortzeko erabiltzen dira.

Linuxen, script-ak hainbat programazio-hizkuntzatan idatz daitezke, hala nola Bash, Python eta Perl, besteak beste. Hala ere, Bash (Bourne Again Shell) script-ak idazteko gehien erabiltzen den programazio-lengoaia da.

Script bat sortzeko, testu lauaren fitxategi bat sortu behar da eta exekuzio-baimenak esleitu behar zaizkio. Adibidez, "myscript.sh" izeneko fitxategi bat sor daiteke eta exekuzio-baimenak esleitu, honako komando honekin:



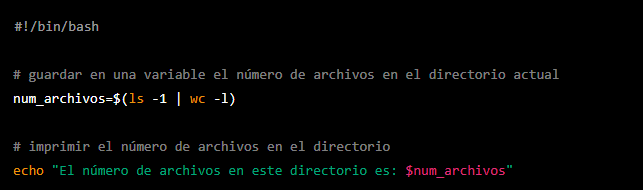
Linuxen script-ek oinarrizko egitura bat dute, goiburu-lerro bat, aldagaiak eta jarraibideak barne hartzen dituena. Goiburu-lerroak erabili beharreko interpretea zehazten du, adibidez, "#!/bin/bash", Bash erabiliko dela adierazteko. Aldagaiak datuak biltzeko eta manipulatzeko erabiltzen diren informazio-edukiontziak dira. Azkenik, aginduak ordenan exekutatuko diren komandoak dira.

Linuxen scriptek fluxua kontrolatzeko funtzio eta egitura batzuk ere ematen dituzte atazak automatizatzen laguntzeko. Adibidez, fluxua kontrolatzeko egiturak erabil daitezke, hala nola "if-else" eta "for", jarraibideen exekuzio-fluxua kontrolatzeko, baldintza jakin batzuen arabera.

Laburbilduz, Linuxen script-ak tresna boteretsua dira atazak automatizatzeko eta programa pertsonalizatuak sortzeko. Hainbat programazio-hizkuntzatan idatz daitezke, baina Bash da gehien erabiltzen dena. Scriptek oinarrizko egitura bat dute, goiburu-lerro bat, aldagaiak eta jarraibideak barne hartzen dituena. Gainera, fluxua kontrolatzeko funtzioak eta egiturak ematen dituzte, jarraibideak gauzatzeko fluxua kontrolatzeko.

Linuxen scripting ariketa erraz bat honako hau izan liteke:

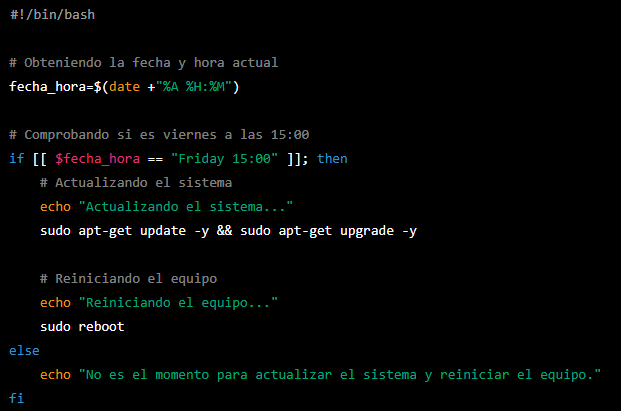
1. Sortu script fitxategi bat, "contar\_archivos.sh" izenekoa, erabiltzailearen home direktorioan.
2. Erantsi kode hau fitxategiari:



1. "chmod +x contar\_archivos.sh" komandoaren script-ari exekuzio-baimenak ematea.
2. Egikaritu script honekin: "./contar\_archivos.sh"

Script-ak fitxategi kopurua kontatzen du script exekutatzen den direktorioan, using "ls -1" command to list files in a directory, and "wc -l" command count lines, then it prints the number of files using echo command.

Sistema eguneratzeko eta ekipoa ostiralero 15: 00etan berrabiarazteko zeregina automatizatzen duen Linux-eko script baten adibidea:



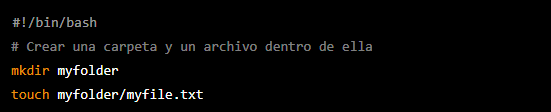
Script honek "date" komandoa erabiltzen du gaur egungo data eta ordua lortzeko, gero "if" egitura erabiltzen du ostirala 15: 00etan den egiaztatzeko. Horrela bada, sistema eguneratzen da "apt-get update" eta "apt-get upgrade" komandoak erabiliz, eta ekipoa berriz hasten da "reboot" komandoa erabiliz.

Zeregin hori automatizatzeko, scripta programatu ahal izango litzateke Linuxen "cron" moduko atazak programatzeko tresna bat erabiliz. Adibidez, scripta ostiralero programatu daiteke 15: 00etan exekutatzeko.

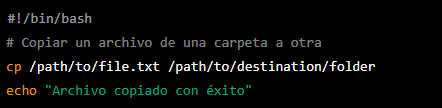


Garrantzitsua da eguneratze- eta berrabiarazte-komandoak exekutatzeko behar diren baimenak badituztela ziurtatzea, eta scripta baimen horiek dituen erabiltzaile batek gauzatzen ari dela ziurtatzea.

Beste *script* adibide batzuk.



"myfolder" izeneko karpeta bat sortzen da, eta fitxategi bat karpeta horren barruan, "myfile.txt" izenekoa, mkdir eta touch komandoen bidez.



"file.txt" izeneko fitxategi bat kopiatzen da ibilbide espezifiko batetik helmugako karpeta batera, cp komandoa erabiliz, eta echo komandoarekin baieztapen-mezu bat erakusten da.

Gogoratu, script-aren izena .sh luzapenarekin bukatu behar dela eta fitxategi horri exekuzio baimena eman behar zaiola.

chmod +x nirescripta.sh

Script-a exekutatzeko: ./nirescripta.sh