

பாடம் 3

லினக்ஸ்

3.1 லினக்ஸின் வரலாறு

லினஸ் டோர்வால்ட்ஸ் (Linus Toruualds), ஃபின்லாந்திலுள்ள ஹெல் சின்க்கி பல்கலைக் கழகத்தில் படித்துக் கொண்டிருந்தபோது, அப்பல் கலைக்கழகத்தின் விதிமுறைகளின்படி அவர் ஒரு திட்டப்பணியை (Project) உருவாக்க வேண்டியிருந்தது. அக்காலகட்டத்தில், பேராசிரியர் ஆண்ட்ரூ எஸ்.டேனன்பாம் (Prof.Andrew S. Tannenbaum) அவர்கள் உருவாக்கிய, யூனிக்ஸின் ஒரு வடிவமான 'மினிக்ஸ்'(Minix) இயக்க முறைமையை மாணவர்கள் பயன்படுத்தி வந்தனர். மினிக்ஸைப் படித்தபின், அதன் தாயான யூனிக்ஸின் நேர்த்தியும் செயல்திறனும் லினஸை வெகுவாகக் கவர்ந்தன. யூனிக்ஸ் இயக்க முறைமை, கென் தாம்சன் (Ken Thompson) அவர்களின் முயற்சியால் உருவாக்கப்பட்டது. மினிக்ஸ் பயனர்களுக்காக, பீசியில் செயல்படக் கூடிய செயல்திறன் மிக்க யூனிக்ஸ் பதிப்பை உருவாக்க லினஸ் டோர்வால்ட்ஸ் தீர்மானித்தார். அந்த இயக்க முறைமைக்குத் தன்னுடைய முதல் பெயரோடு யூனிக்ஸின் (மினிக்ஸின் எனவும் கொள்ளலாம்) கடைசி எழுத்தைச் சேர்த்து 'லினக்ஸ்' (Linux) எனப் பெயரிட்டார். 1991 ஆம் ஆண்டில் அதன் முதல் பதிப்பு 0.11-ஐ வெளியிட்டார். லினக்ஸ், இணையம் வழியாகப் பரவலாக விநியோகிக்கப்பட்டது. தொடர்ந்த ஆண்டுகளில் நிரலர்கள் பலர் புதிய இயக்க முறைமையான லினக்ஸைச் செழுமைப்படுத்தினர். யூனிக்ஸ் முறைமைகளில் காணப்பட்ட சில பண்புக் கூறுகளைச் சேர்த்தனர். உலகம் முழுவதிலுமுள்ள ஆர்வம் மிக்க நிரலர்கள் லினக்ஸின் மேம்பாட்டுக்காக அவரவர் பங்களிப்பைச் செலுத்தினர். இந்த வகையில், லினக்ஸ், பிற இயக்க முறைமைகளைவிடத் தனித்த அனுகூலம் பெற்றுள்ளது. பொதுவாக, இயக்க முறைமைகள் கட்டுத்திட்டமான சூழ்நிலையில் உருவாக்கப்படுகின்றன. முழுக் குறிமுறையிலும் (entire code) காணப்படும் பிழைகளைக் குறிப்பிட்ட காலக்கெடுவுக்குள் களைவதற்கு, வரம்புக்குட்பட்ட எண்ணிக்கையிலான நிரலர்களே அனுமதிக்கப்படுவர். ஆனால் லினக்ஸ் குறிமுறையோ அனைவருக்கும் கிடைக்கிறது. அவர்கள், எவ்விதக் கட்டுப்பாடும் இன்றித் தோழமையான சூழலில் அதிலுள்ள பிழைகளைக் கண்டறிந்து திருத்த முடியும்.

லினக்ஸ், அதன் மிக எளிய வடிவத்தில், இயங்குவதற்கு வெறுமனே 4 எம்பி நினைவகம் மட்டுமே இருந்தால் போதும். இந்த இயக்க முறைமை அதன் அனைத்து வசதிகளோடும் இந்த அளவுக்குக் குறைந்த நினைவகம்

கத்தையே எடுத்துக்கொள்கிறது என்பது வியத்தகு செய்தியாகும். இதன் காரணமாக, லினக்ஸின் நிலைப்புறுதியோ (Stability) வேகமோ (Speed) பாதிக்கப்படவில்லை.

உலகம் முழுவதிலுமுள்ள எத்தனையோ நிரலர்களின் பங்களிப்பில் லினக்ஸ் உருவாக்கப்பட்டபோதும், அதன் வடிவம் சிதைந்து போகவில்லை. ஒரு கட்டுக்கோப்புக்குள்ளேயே இருந்து வருகிறது. பிறப்பு காலம் தொட்டு லினக்ஸ், யூனிக்ஸுக்கான அன்சி (ANSI) தரப்பாடாகக் கருதப்படும் போசிக்ஸ் (POSIX- Portable Operating System Interface for Computer Environments) வரைமுறைகளுக்கு உட்பட்டே உருவாக்கப்பட்டது. குறிப்பாக, இன்டெல் அடிப்படையிலான பீசிகளுக்கென்றே லினக்ஸ் வடிவமைக்கப்பட்டுள்ளது.

லினக்ஸின் ஒவ்வொரு வெளியீட்டுக்கும் இரண்டு பதிப்புகள் உள்ளன. அவற்றுள் ஒன்று நிலைத்த பதிப்பு (Stable version) மற்றது பரிசோதனைப் (beta) பதிப்பாகும். n.x.y என்னும் பதிப்பெண்ணில், முதல் எண் n, பதிப்பு எண்ணைக் குறிக்கிறது. இரண்டாம் எண் x, இரட்டைப்படையாக இருப்பின் அது நிலைத்த பதிப்பு, இல்லையேல் அது பரிசோதனைப் பதிப்பு. எடுத்துக்காட்டாக, 2.2.25 என்பதில் x=2 ஆகும். எனவே இது ஒரு நிலைத்த பதிப்பு.

லினக்ஸின் வளர்ச்சிக்கு இணையம் ஒரு வரமாக அமைந்துள்ளது; உலகம் எங்கிலும் உள்ள கணிப்பொறி வல்லுநர்கள் லினக்ஸை வளர்த்தெடுக்கும் பொருட்டுப் பிறரோடு இணைந்து பணியாற்ற இணையம் வழிவகுத்துள்ளது. இன்றைக்குப் பல நிறுவனங்கள் இணையம் வழியாக லினக்ஸுக்கு ஆதரவு நல்கி வருகின்றன. இணையத்தில் ஏராளமான லினக்ஸ் குழுக்கள் செயல்பட்டு வருகின்றன. இந்த அமைப்புகளில் பதிவு செய்து கொள்ளக் கட்டணம் எதுவுமில்லை. நீங்கள் உறுப்பினராகி, இந்தக் குழுக்களிடமிருந்து லினக்ஸ் பற்றிய மிக அண்மையத் (latest) தகவல்களைப் பெறமுடியும்.

3.2 லினக்ஸில் உள்புகுதலும் வெளியேறுதலும் (Logging in / Logging out of Linux)

லினக்ஸ் முறைமை பல பயனர்கள் ஒரே நேரத்தில் பணியாற்ற அனுமதிக்கிறது. பயனர் பெரும்பாலும் பயனர் முனையத்தில் (user terminal) பணியாற்றுவார். முதலில் நீங்கள் லினக்ஸ் முறைமையோடு இணைப்பை ஏற்படுத்திக் கொள்ளவேண்டும். லினக்ஸ் முறைமை சில தகவல்களைக் காட்டி முடிந்தபின், உள்புகு தூண்டியைக் (prompt) காட்டும். ('தூண்டி'

என்பது தூண்டுகுறி அல்லது தூண்டு செய்தியாகும்). அங்கே உங்களின் உள்புகு பெயரை (login name) உள்ளிட வேண்டும். உள்புகு பெயர் என்பது உங்களுக்குரிய பயனர் பெயர் (username) ஆகும். 'லாகின்' (Login) என்ற சொல்லைக் கண்டு அஞ்சத் தேவையில்லை. இங்கே 'பயனர்' என்ற பொருளைத் தருகிறது, அவ்வளவுதான்!

ஒரு வீட்டில் பல அறைகள் உள்ளன, ஓர் அறையில் ஒருநபர் எனத் தங்கியுள்ளனர். ஒவ்வோர் அறையிலும் மதிப்புமிக்க புதையல் இருப்பதாகக் கொள்வோம். அந்த வீட்டிலுள்ள ஒவ்வோர் அறையையும் காத்துக் கொள்வது எப்படி? வீட்டுக்குள் நுழைவதற்கான ஒரே பொது வானகதவு உட்புறம் தாளிடப்பட்டுள்ளது. காவலர் ஒருவர் நியமிக்கப்பட வேண்டும். நீங்கள் வீட்டுக்குள் நுழைய வேண்டும் எனில் அவரிடம் உங்கள் பெயரையும் உங்களுக்குரிய ரகசிய அடையாளக் குறியீட்டையும் கூற வேண்டும். வீட்டில் குடியிருப்போர் அனைவரின் பெயர்கள் மற்றும் பெயர்களோடு தொடர்புடைய அடையாளக் குறியீடு ஆகியவை அடங்கிய பட்டியல் ஒன்றை வைத்திருப்பார். நீங்கள் கூறும் பெயரும் அடையாளக் குறியீடும் அந்தப் பட்டியலில் இருக்குமெனில், இருந்தால் மட்டுமே, உங்கள் அறைக்குள் நுழைய அனுமதி தருவார். பட்டியலில் இல்லை எனில், உங்கள் பெயரையும் அடையாளக் குறியீட்டையும் திரும்பவும் கூறும்படி சொல்வார். நீங்கள் சரியாகச் சொல்லும்வரை இவ்வாறு தான் சொல்வார். உள்ளே அனுமதிக்கமாட்டார்.

லினக்ஸிலும் இவ்வாறே நிகழ்கிறது. லினக்ஸ் முறைமையுடன் இணைத்துக்கொண்டதும், உள்புகு தூண்டு செய்தியில் (Login prompt) உங்களின் உள்புகு பெயரை (Login name) உள்ளிடுமாறு கேட்டுக் கொள்ளப் படுவீர்கள். உங்களுக்குரிய உள்புகு பெயரை உள்ளிடவேண்டும். அடுத்து, கடவுச்சொல் தூண்டுசெய்தி (Password prompt) காட்டப்படும். உங்களுக்குரிய கடவுச்சொல்லை உள்ளிடுங்கள். உங்கள் கடவுச் சொல் ரகசியமாகப் பாதுகாக்கப்படவேண்டும். எனவே, லினக்ஸ் முறைமை, நீங்கள் தரும் கடவுச்சொல்லை திரையில் காட்டாது. உங்கள் பெயரும் கடவுச் சொல்லும் முறைமைக் கோப்புகளிலுள்ள விவரங்களுடன் ஒப்பிடப் படும். பொருந்தும் விவரம் காணப்படவில்லை எனில், இந்த இரண்டு விவரங்களையும் மீண்டும் தரவேண்டும். அதே வேளையில், பொருந்தும் விவரம் காணப்படின், கீழேயுள்ளதுபோன்ற செய்தி காட்டப்படும்.

[ilamathi@localhost ilamathi]\$

இதில் இளமதி என்பது, உங்கள் உள்புகு பெயர் எனக் கருதிக் கொள்க. சிறிய எழுத்து (lowercase) மற்றும் பெரிய எழுத்து (upper case)

களில் மிகவும் கவனமாக இருக்கவேண்டும். காரணம் லினக்ஸ் முறைமை எழுத்து வடிவ உணர்வுள்ளது (case sensitive). அனுமதியில்லாத நபர்கள் உங்கள் கோப்பகங்கள் (directories) அல்லது கோப்புகளை (files) அணுக முடியாது. பொதுவாக, ஒரு பயனரின் பெயரே அவரின் உள்புகு பெயராக இருக்கவேண்டிய தேவையில்லை. ஒரு பயனர் முதன் முறையாக முறைமையினுள் நுழையும்போது, அவருக்குரிய இல்லக் கோப்பகம் (Home Directory) முறைமை நிர்வாகியால் (System Administrator - SA) ஒதுக்கப்படுகிறது. எஸ்ஏ பற்றிப் பிறகு கற்பீர்கள்.

வெளியேறு செயல்முறை (Logout Process)

உங்கள் இல்லக் கோப்பகத்தை மூடாமல் முறைமையை விட்டு வெளியே வந்தால், மற்றவர்கள் உங்கள் தகவல்களுக்குத் தீங்கு விளைவிக்கக்கூடும். எனவே, லினக்ஸ் முறைமையைவிட்டு முறைப்படி வெளியேறுவது (logout) கட்டாயமாகும். கட்டளைத் தூண்டுகுறியில் (Command Prompt) exit அல்லது logout என உள்ளிட்டால், உங்களின் நடப்பு லினக்ஸ் அமர்வு முடிவுக்கு வரும். அடுத்து, முறைமை, பிற பயனர்களுக்காக உள்புகு தூண்டுகுறியை திரையில் காட்டும்.

கடவுச்சொல்லை மாற்றுதல் (Changing the password)

யாரோ ஒருவர் உங்கள் கடவுச்சொல்லை அறிந்துவிட்டாரோ என்கிற மன உலைவு உங்களுக்கு ஏற்படுவதாக வைத்துக்கொள்வோம். பிறகு, இரவெல்லாம் தூக்கமில்லாமல் தவிக்க நேரிடும். லினக்ஸ் இயக்க முறைமை, உங்கள் சிக்கலைத் தீர்ப்பதற்குச் சிறந்த வழிமுறையை வழங்குகிறது. உங்கள் கடவுச் சொல்லை மாற்றிக்கொள்ள அனுமதிக்கிறது. passwd என்னும் கட்டளை மூலம் ஒரு பயனர் அவருடைய கடவுச்சொல்லை மாற்றிக் கொள்ளலாம். இளமதி என்னும் பயனர் தன்னுடைய கடவுச் சொல்லை மாற்றிக்கொள்ளப் பின்பற்றிய படிநிலைகள் கீழே தரப்பட்டுள்ளன. உங்கள் கணிப்பொறியில் காண்பது இதையொத்தே இருக்கும்.

எடுத்துக்காட்டு:

```
[ilamathi@localhost ilamathi]$ passwd
```

Changing password for ilamathi (current) password:

(பயனர் தன் தற்போதைய கடவுச்சொல்லை உள்ளிடுகிறார்)

New password: (பயனர் புதிய கடவுச்சொல்லை உள்ளிடுகிறார்)

Retype new password:

(பயனர் புதிய கடவுச்சொல்லை மீண்டும் உள்ளிடுகிறார்)

passwd: all authentication tokens updated successfully

[ilamathi@localhost ilamathi]\$ _

passwd கட்டளை பழைய கடவுச்சொல்லைக் கேட்கும். பயனரின் சான்றுதியைச் சரிபார்க்க இந்தக் கட்டளை அவசியம் ஆகும். இல்லை யேல் தீங்கெண்ணம் கொண்டோர் பிற பயனர்களின் பணியில் பேரழிவுகளை ஏற்படுத்தமுடியும். புதிய கடவுச்சொல்லைப் பற்றிப் பயனருக்கு தீர்க்கமான முடிவிருக்கவேண்டும் என முறைமை எதிர்பார்க்கிறது. எனவேதான் புதிய கடவுச்சொல்லை இருமுறை தரச்சொல்லிக் கேட்கிறது.

எடுத்துக்காட்டு:

[ilamathi@localhost ilamathi]\$ passwd

Changing password for ilamathi

(current) password:

(பயனர் தற்போதைய கடவுச்சொல்லை உள்ளிடுகிறார்)

New passwd:(பயனர் புதிய கடவுச்சொல்லை உள்ளிடுகிறார்)

Retype new password: (பயனர் பிழையான கடவுச்சொல்லை தருகிறார்)

Sorry, passwords do not match

New password:

தற்போதைய கடவுச்சொல்லை (நீங்கள் மாற்ற விரும்பும் கடவுச்சொல்) உள்ளீடு செய்யும்போது, தவறு இழைப்பின், நீங்கள் சான்றுதி பெற்ற பயனர் அல்ல என்றாகிவிடும். எனவே, நீங்கள் வேறு ஒருவரின் கடவுச்சொல்லை மாற்ற முயல்கிறீர்கள் என்கிற ஐயம் ஏற்பட்டுவிடும். எனவே, கடவுச்சொல்லை மாற்ற உங்களுக்கு அனுமதி வழங்கப்பட மாட்டாது. இருமுறை தருகின்ற புதிய கடவுச்சொல் ஒத்துப்போகவில்லை எனில், முறைமை குழப்பமடைந்து விடும். எந்தக் கடவுச் சொல்லை ஏற்றுக்கொள்வது என்பதை முறைமையால் தீர்மானிக்க முடியாது. நீங்கள் தரும் புதிய கடவுச்சொல் ஒரேயடியாய்த் தள்ளுபடி செய்யப்படும். எனவே, பழைய கடவுச்சொல் மட்டுமன்றி புதிய கடவுச்சொல்லையும் கவனத்துடன் தரவேண்டும்.

கடவுச்சொல்லை மாற்றுவது தொடர்பான விதிமுறைகளுள் சில:

- ✱ கடவுச்சொல் ஆறு எழுத்துகளுக்கும் குறைவான நீளத்துடன் இருக்கக்கூடாது.
- ✱ கடவுச்சொல்லை மாற்றும்போது, புதிய கடவுச்சொல் பழைய

கடவுச் சொல்லிலிருந்து மூன்று எழுத்துகளாவது மாறுபட்டிருக்க வேண்டும்.

- ✱ கடவுச்சொல் பயனரின் உள்பகு பெயரிலிருந்து வேறுபட்டிருக்க வேண்டும்.
- ✱ புதிய கடவுச்சொல்லும் பழைய கடவுச்சொல்லும் ஒன்றாக இருக்க முடியாது. ஒரு கடவுச்சொல்லை அதே சொல்லால் மாற்ற முனைவது நகைப்பிற்குரியது அல்லவா?
- ✱ முறைமை நிர்வாகியானவர் முறைமை முழுவதின் மீதும் கட்டுப்பாடு செலுத்த வல்லவர். எனவே, லினக்ஸ் முறைமையில் அவர் எந்தப் பயனரின் கடவுச் சொல்லையும் மாற்றியமைக்கமுடியும்.

3.3 லினக்ஸ் கோப்பு முறைமை (The Linux File System)

தகவல்களை இழந்துவிடாமல் தக்கவைத்துக்கொள்ள விரும்புகிறீர்களா? அப்படியெனில் அவற்றைக் கணிப்பொறியில் எவ்வாறு சேமிப்பீர்கள்? தகவலை ஒரு கோப்பில் சேமித்து வைக்க முடியும். ஆனால், கோப்பு என்பது யாது?

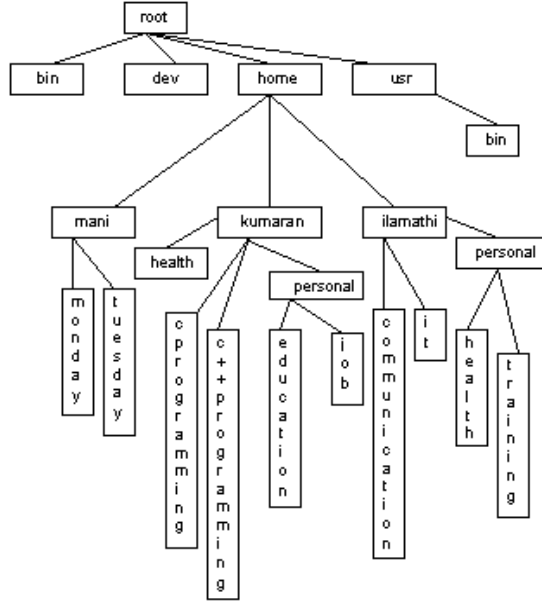
கோப்பு என்பது ஏடுகளின் தொகுப்பு. ஏடு என்பது ஓர் உருபொருள் (entity) பற்றிய தகவலைத் தருகிறது. உருபொருள் என்பது, ஒரு மாணவராக இருக்கலாம் அல்லது பயணம் செய்யும் ஒரு ரயில் பயணியாக இருக்கலாம். உங்களின் மதிப்பெண் தாளும் ஓர் ஏடுதான். ஏடு என்பது புலங்களைக் கொண்டிருக்கும். மிக அடிப்படையான புலங்களைப் பொருள் விளங்குமாறு மேலும் சிறு புலங்களாகப் பிரிக்க முடியாது. பெயர், வரிசை எண் ஆகியவை புலங்களுக்கு எடுத்துக்காட்டுகளாகும். உங்கள் வகுப்புத் தோழர்கள் அனைவரின் மதிப்பெண் தாள்களின் தொகுப்பு, கோப்புக்கு எடுத்துக்காட்டாகும்.

உங்கள் பள்ளியில் உள்ள அனைத்து மாணவர்களின் மதிப்பெண் தாள்களின் தொகுப்பை எடுத்துக்கொள்வோம். இது கோப்புகளின் தொகுப்பாகும். இது கோப்பகம் (Directory) என்று அழைக்கப்படுகிறது. மதிப்பெண் தாளைப் போன்று பல கோப்புகள் உங்கள் பள்ளியில் இருக்க முடியும். எடுத்துக்காட்டாக, மாற்றுச் சான்றிதழ் கோப்பு, சம்பளக் கணக்குக் கோப்பு, இன்னும் பிற. அவை தனியே தலைப்பிட்டு சேமிக்கப்பட வேண்டும். மாற்றுச் சான்றிதழுக்காக, ஒவ்வொரு மாணவரைப் பற்றியும் பள்ளியில் தகவல் இருக்கும். ஒவ்வொரு வகுப்புக்கும் உரிய தகவல் ஒரு கோப்பாக இருக்கும்.

ஓரே வகையைச் சார்ந்த கோப்புகளின் தொகுப்பு ஏற்கெனவே கூறிய

படி 'கோப்பகம்' எனப்படும். இந்த வகையில், ஒரு பள்ளியில் பல்வேறு தொகுதிகளாகக் கோப்புகள் இருக்க முடியும். மதிப்பெண் தாள்களைக் கொண்ட கோப்பு, மாற்றுச் சான்றிதழுக்கான தகவல் கோப்பிலிருந்து பிரிக்கப்படவேண்டும். எனவே, கோப்புகளின் பிற தொகுதிகளிலிருந்து மதிப்பெண் தாள் கோப்புகளை பள்ளி நிர்வாகம் தனியே பராமரிக்க வேண்டும். ஒரே வகையான கோப்புகளின் தொகுதி ஒரு கோப்பகத்தில் இருத்தப்படவேண்டும். ஒரு நிலைவட்டில் ஆயிரக்கணக்கான கோப்புகளைச் சேமிக்க முடியும். மேலே கூறியுள்ளவாறு கோப்புகளை ஒழுங்கமைப்பின், குறிப்பிட்ட கோப்பினைத் தேடிக் கண்டறிவது எளிதாக இருக்கும்.

அனைத்துக் கோப்புகளும் வட்டினில், 'மூலக் கோப்பகம்'(root directory) எனப்படும் முதன்மையான கோப்பகத்தின் கீழ் சேமிக்கப்படுகின்றன. மரம் போன்ற கட்டமைப்பில் கோப்புகள் ஒழுங்கமைக்கப்படுகின்றன. தலைகீழாக நிற்குகொண்டு ஒரு (அடிமரம் இல்லாத) மரத்தைப் பாருங்கள். மரத்தின் வேர் உச்சியில் இருக்கும். அதிலிருந்து கிளைகள் படர்ந்திருக்கும். கிளைகளில் சிறு சிறு கிளைகள் கிளைத்திருக்கும். இறுதியில் இலைகள் இருக்கும். இலைகள் கோப்புகளை உருவகப்படுத்துகின்றன. கிளைகள், கோப்பகங்களையும் (directories) உள்-கோப்பகங்களையும் (sub-directories) உருவகப்படுத்துகின்றன. மரத்தின் வேர்தான் மூலக் கோப்பகம் (root directory) என்பதை நீங்களே ஊகித்திருப்பீர்கள்.



படம் 3.1 லினக்ஸ் கோப்பகக் கட்டமைப்பு

மூலக் (root) கோப்பகம், bin, boot, home, usr, etc, lib, dev, tmp போன்ற கோப்பகங்களாகப் பிரிக்கப்படுகின்றன. இல்லக் கோப்பகம் (home directory) /home எனக் குறிக்கப்படுகிறது. இல்லக் கோப்பகம் என்பது / (root) என்னும் மூலக்கோப்பகத்தின் 'சேய்' என்பதை உணர்த்துகிறது. மூலக் கோப்பகம், இல்லக் கோப்பகத்தின் 'தாய்' ஆகும். மூலக் கோப்பகத்துக்குத் 'தாய்' கிடையாது.

ஒரு குழந்தை பிறந்தவுடன் அதற்குப் பெயர் சூட்டவேண்டும். அது போல ஒரு கோப்பினை உருவாக்கும்போது அதற்குப் பெயரிடவேண்டும். ஒரு கோப்பினை உருவாக்கும்போது கீழே காணும் விதிமுறைகளைப் பின்பற்றவேண்டும்.

கோப்பின் பெயர்-

1. எழுத்துகள், அடிக் கீறுகள் (underscore), எண்கள், புள்ளிகள், காற்புள்ளிகளைக் கொண்டிருக்கலாம்.
2. 256 எழுத்துகள் வரை இருக்கலாம்.
3. முதலெழுத்து, எண்ணாக இருக்கக்கூடாது.
4. புள்ளியுடன் தொடங்கக்கூடாது.

(சில குறிப்பிட்ட சூழ்நிலைகளில், புள்ளியுடன் தொடங்கும் பெயரை ஒரு கோப்புக்குச் சூட்டலாம் என்ற போதிலும், அவ்வாறு சூட்டாமல் இருப்பதே நல்லது. புள்ளியுடன் தொடங்கும் பெயர்கொண்ட கோப்பு புள்ளிக் கோப்பு (dot file) என்றழைக்கப்படுகிறது. புள்ளிக் கோப்புகள் பெரும்பாலும் முறைமை பயன்படுத்துகிற கோப்புகளாகும்).

5. சாய்வுக் கோடு (Slash), கேள்விக்குறி, நட்சத்திரக்குறி இடம்பெறக்கூடாது.
6. கட்டளைச் சொல்லாக இருக்கக்கூடாது.

கோப்பின் பெயர் ஒரு முதன்மைப் பெயர் (Primary name) , ஒரு துணைநிலைப் பெயரைக் (Secondary Name) கொண்டிருக்கலாம். புள்ளிக்கு முந்தைய பெயர் முதன்மைப் பெயராகும். புள்ளிக்குப் பிந்தைய பெயர் துணைநிலைப் பெயர் அல்லது நீட்டிப்பு (extension) எனப்படுகிறது. கோப்புகளை வகைப்படுத்துவதற்கு நீட்டிப்பு பயன்படுகிறது. எடுத்துக்காட்டாக, C என்ற நீட்டிப்புக் கொண்ட கோப்புகளை (C கோப்புகள்) எடுத்துக்கொள்வோம். அவை சி-மொழியில் எழுதப்பட்ட நிரல் கோப்புகளைக் குறிக்கின்றன.

ஒரு கோப்பு அல்லது கோப்பகத்தை உருவாக்கும்போது நீங்கள் சூட்டும் பெயர் அதன் முழுப்பெயர் ஆகாது. ஒரு கோப்பு அல்லது கோப்பகத்தின் முழுப்பெயர் என்பது அதன் முழுப் பாதையையும் உள்ளடக்கியதாகும். ஒரு கோப்பு அல்லது கோப்பகத்தைக் குழப்பமின்றி அடையாளங்காட்ட, கோப்பு முறைமையின் மரக்கட்டமைப்பைப் பயன்படுத்திக்கொள்ள முடியும்.

எடுத்துக்காட்டாக, `cprogramming` என்ற கோப்பின் முழுப்பெயர்,

`/home/kumaran/cprogramming`

ஆகும்.

3.4 பயனர்களின் வகைகள்

லினக்ஸ் முறைமையில் நான்கு வகையான பயனர்கள் உள்ளனர்.

1. முறைமை நிர்வாகி அல்லது மூலப் பயனர் (System Administrator or the Root user)

முறைமையின் சுமுகமான செயல்பாட்டுக்கு முறைமை நிர்வாகியே (எஸ்ஏ) முதன்மையாகப் பொறுப்பேற்கிறார். எஸ்ஏ, பயனர்களின் `/home` கோப்பகங்களை உருவாக்குகிறார். பயனர்களின் குழுக்களுக்கு முறைமையின் சார்பாகச் சேவை புரிகிறார். கணிப்பொறியில் நெகிழ்வட்டு மற்றும் சிடி-ரோம் வட்டுகளை இவர்மட்டுமே பயன்படுத்த முடியும். கணிப்பொறி செயலிழந்து போகும்போது ஏற்படும் தகவல் இழப்பு களைத் தவிர்க்கக் காப்புநகல்களை இவர் எடுக்கிறார். லினக்ஸில் இவர், மூலப் பயனர் (root user) அல்லது மீப் பயனர் (super user) என்று அழைக்கப்படுகிறார். மூலப் பயனருக்கான தூண்டுகுறி (prompt), # ஆகும். பிற பயனர்களுக்கு \$ ஆகும்.

2. கோப்பு உரிமையாளர் (File Owner)

ஒரு கோப்பினை உருவாக்கும் பயனர் அந்தக் கோப்பின் உரிமையாளர் எனப்படுகிறார். ஒரு கோப்பின் உரிமையாளர், அந்தக் கோப்பினை நகலெடுத்தல், அழித்தல், திருத்துதல் போன்ற எந்தச் செயல்பாட்டையும் மேற்கொள்ள முடியும். `chmod` கட்டளை மூலம் கோப்பின் அணுகு அனுமதியை மாற்றியமைத்து, அதனை இயக்க (execute) முடியும். `chmod` கட்டளைபற்றிப் பிறகு படிப்பீர்கள்.

3. குழு உரிமையாளர் (Group Owner)

ஒரு குறிப்பிட்ட திட்டப்பணியில் பணியாற்றிக்கொண்டிருக்கும் குழு

வினர் பயனுறுதி கருதி அவர்களுடைய கோப்புகளைப் பகிர்ந்துக்கொள்ள வேண்டும். இந்தக் கோப்புகள் குழுத் தலைவரின் /home கோப்பகத்தில் உருவாக்கப்படுகின்றன. அக்குழுவின் அனைத்து உறுப்பினர்களும் தம் கோப்புகளைப் பகிர்ந்து கொள்கிறார்கள். இந்தக் குழுவினர் குழுப் பயனர்கள்(group users) என்று அழைக்கப்படுகின்றன. ஒரு பயனருக்குப்பெயர் இருப்பது போல, பயனர்களின் குழுவுக்கும் ஒரு பெயர் சூட்டப்படுகிறது.

4. பிற பயனர்கள் (Other users)

முறைமையில் பணியாற்றுகிற, ஒரு திட்டப்பணிக் குழுவில் உறுப்பினராக இல்லாத பயனர்கள் அனைவரும் அக்குழுவின் கோப்புகளைப் பொறுத்தமட்டில் 'பிற பயனர்கள்' என்றே குறிக்கப்படுகின்றனர். குறிப்பிட்ட குழுவில் உறுப்பினரல்லாத பயனர்கள் அக்குழுவைப் பொறுத்தமட்டில் 'பிற பயனர்கள்' ஆவர்.

3.5 கோப்பகக் கட்டளைகள் (Directory Commands)

கட்டளைகளின் பொதுவடிவம்

கட்டளைகளைப் பற்றிக் கற்பதற்கு முன்பாக, கட்டளைகள் பற்றிய சில அடிப்படையான கருத்துருக்களை (Concepts) நீங்கள் அறிந்துகொள்ள வேண்டும். ஒரு கட்டளையின் பொதுவான வடிவமைப்பு இது:

கட்டளை	[தேர்வு]	[செயலுருபு]
(Command)	[option]	[argument]

எல்லாக் கட்டளைகளுக்கும் தேர்வு (option), செயலுருபு (argument) இரண்டும் இருக்க வேண்டும் என்கிற கட்டாயமில்லை. சில கட்டளைகளுக்குத் தேர்வுமட்டும் தேவைப்படலாம். வேறு சிலவற்றுக்குச் செயலுருபு மட்டும் தேவைப்படலாம். இன்னும் சிலவற்றுக்கு இரண்டுமே தேவைப்படலாம். எடுத்துக்காட்டாக, **ls** கட்டளை, தேர்வு, செயலுருபு இரண்டும் இல்லாமலும் செயல்படும். தேர்வு மட்டும் இருந்தாலும், செயலுருபு மட்டும் இருந்தாலும், இரண்டுமே இருந்தாலும் செயல்படும். [**ls** என்பது list என்பதன் சுருக்கம். ஆனால், **ls** என்பதற்குப் பதிலாக, கணிப்பொறிக்குப் புரியாதுமே என்ற எண்ணத்தில் list எனத் தரக்கூடாது. உங்கள் நல்லெண்ணத்துக்குப் பிழைசுட்டும் செய்தியே (error message) பரிசாகக் கிடைக்கும். **ls** கட்டளை மேலும் இரண்டு எழுத்துகள் விசை பதிவு (typing) செய்வதை மிச்சப்படுத்துகிறது. அதே வேளையில் உங்கள் நினைவகத்துக்குச் சுமை ஏற்றுகிறது].

எடுத்துக்காட்டு:

கட்டளை	தேர்வு	செயலுருபு
ls		
ls	-l	
ls		*.C
ls	-l	file1

இக்கட்டளை பற்றிப் பின்னால் விரிவாகப் படிக்க இருக்கிறோம்.

கட்டளை என்பது ஒற்றைச் சொல்லால் ஆனது. கட்டளை பொதுவாக ஓர் எழுத்தில் தொடங்கும். கட்டளைச் சொற்களைப் பொறுத்தவரை உங்களுக்குச் சிறப்புச் சலுகை எதுவும் கிடையாது. சொல்லின் எழுத்தாக்கத்தையும் (spelling), எழுத்துகளின் வடிவத்தையும்(case) உள்ளது உள்ளபடியே தரவேண்டும்.

கட்டளையில் தேர்வு (option) என்பது ஒரு கழித்தல் (dash) குறியுடன் தொடங்கும். அதைத் தொடர்ந்து ஒரேழுத்து இடம்பெறும். ஆனால் ஒன்றுக்கு மேற்பட்ட தேர்வுகளைச் சேர்த்து ஒன்றாகக் குறிப்பிடலாம். எழுத்தின் வடிவம்(case) மிகவும் முக்கியமானது. கழித்தல் குறிக்கும் அடுத்து வரும் ஒரேழுத்துக்கும் இடையே வெற்று இடவெளி (Blank space) இருக்கக் கூடாது. கழித்தல் குறியே, கட்டளை மற்றும் செயலுருபிலிருந்து வேறுபடுத்தித் 'தேர்வு'(option) என உணர்த்துகிறது. ஒரே தேர்வு வெவ்வேறு கட்டளைகளில் ஒரே மாதிரியாகச் செயல்படும் என எதிர் பார்க்கமுடியாது. தேர்வுக்கு முன் நிற்கும் கட்டளைச் சொல்லே தேர்வின் மீது தாக்கம் செலுத்துகிறது. எடுத்துக்காட்டாக, -a என்னும் தேர்வு ls கட்டளையில் (ls -a), மறைக்கப்பட்ட கோப்புகள் உட்பட நடப்புக் கோப்பகத்திலுள்ள அனைத்துக் கோப்புகளின் விவரங்களையும் எடுத்துக்காட்டும். ஆனால், அதே -a என்கிற தேர்வு [mark -ge 81 -a mark -le 100] என்னும் கட்டளையில் 'தருக்கமுறை உம்' (logical and) என்னும் பொருளைத் தருகிறது.

பொதுவாக, செயலுருபு என்பது, எந்தக் கோப்பின் மீது கட்டளை செயல்படுத்தப்படுகிறதோ அந்தக் கோப்பின் பெயராக இருக்கும். அது எப்போதும் ஓர் எழுத்துடன் (alphabet) தொடங்கும். 'புள்ளிக் கோப்புகள்' (dot files) என அழைக்கப்படும் சில முறைமைக் கோப்புகளின் பெயர்கள் புள்ளியில் (.) தொடங்குவதுண்டு.

படம் 3.1-ல் இரண்டு bin மற்றும் இரண்டு personal கோப்பகங்கள் இருந்தபோதிலும், அவற்றின் பாதைப் பெயர் (path name) காரணமாக,

முறைமை, அவற்றைத் தனித்து அடையாளம் காணும். **bin** கோப்பகங்களின் பாதைப் பெயர்கள் **/bin** மற்றும் **/usr/bin** ஆகும். முதல் சாய்வுக் கோடு எப்போதும் மூலக் கோப்பகத்தை உணர்த்துகிறது. அதுபோலவே, **personal** கோப்பகங்களின் பாதைப் பெயர்கள் **/home/kumaran/personal** மற்றும் **/home/ilamathi/personal** ஆகும். **/home** வரையுள்ள பாதை இரண்டு **personal** கோப்பகங்களுக்கும் பொதுவாக இருந்தபோதிலும், அதன்பிறகு இரண்டும் வெவ்வேறு பாதையைக் கொண்டுள்ளன. இத்தகைய மரவுருக் கட்டமைப்பு (**tree structure**) பெயர்களுக்கு இடையேயான மோதலைத் தவிர்க்கின்றன.

அனைத்துக் கோப்புகளையும் ஒரே இடத்தில் சேமித்து வைத்தால், ஒரே பெயரை கோப்புகளுக்குப் பயன்படுத்த முடியாது. மேலும் ஒரு குறிப்பிட்ட கோப்பினைத் தேடிக் கண்டறிய அதிக நேரம் பிடிக்கும். லினக்ஸில் பாதையின் பெயரைக் குறிப்பிடும்போது, அப்பாதையில் இடம் பெறும் கோப்பகங்களை முன்சாய்வுக்கோடு (**Forward slash**) மூலம் பிரிக்க வேண்டும். ஏற்கெனவே கூறியபடி, பாதையில் முதல் கோப்பகத்தின் முன்னிருக்கும் சாய்வுக்கோடு மூலக் கோப்பகத்தைக் குறிக்கிறது. பாதைப் பெயர்கள் கோப்புகளுக்கும் பொருந்துகின்றன. ஒரு கோப்பானது, ஒரு கோப்பகத்துக்குள், ஒரு பெயர் குறிப்பிட்டு உருவாக்கப்படுகிறது. கோப்பின் பெயரோடு மூலக் கோப்பகம் தொடங்கி, அக்கோப்பு இருக்கும் கோப்பகம் வரையிலான அனைத்துக் கோப்பகங்களின் பாதையோடு சேர்த்தே, முறைமையானது. அக்கோப்பினை அடையாளம் காண்கிறது.

பாதைப் பெயர்கள் இரு வகைப்படும். அவை:

1. முற்றுநிலைப் பாதைப்பெயர் (**Absolute Path name**)
2. சார்புநிலைப் பாதைப்பெயர் (**Relative Path name**)

மூலக் கோப்பகத்துடன் தொடங்கும், ஒரு கோப்பு அல்லது கோப்பகத்தின் முழுமையான பாதைப் பெயர், முற்றுநிலைப் பாதைப்பெயர் எனப்படுகிறது.

சார்புநிலைப் பாதைப்பெயர் நீங்கள் தற்போது பணிபுரியும் கோப்பகத்தில் தொடங்கும். நீங்கள் பணியாற்றும் கோப்பகத்தைச் சார்ந்து, அதன் பிறகு அமையும் பாதையை மட்டும் குறிக்கிறது.

படம் 3.1-ல் காணும் கோப்பகக் கட்டமைப்பின்படி, நீங்கள் பணிபுரியும் கோப்பகம் **kumaran** எனில், **job** என்னும் கோப்பின் சார்புநிலைப் பாதைப் பெயர் **/personal/job** ஆகும். அதே கோப்பின் முற்றுநிலைப் பாதைப் பெயர் **/home/kumaran/personal/job** ஆகும்.

நீங்கள் தற்போது பணிபுரிந்து கொண்டிருக்கும் நடப்புக் கோப்பகப் பாதையை எவ்வாறு கண்டறிவீர்கள்? சில வேளைகளில், கோப்பக அமைப்பில், நீங்கள் எந்தக் கோப்பகத்தில் பணிபுரிகிறீர்கள் என்பது தெரியாமல் போகலாம். அப்படிப்பட்ட சூழ்நிலைகளில் pwd கட்டளை (print working directory or path of working directory or present working directory) கைகொடுக்கும். இளமதி என்பவர், முறைமையில் உள்புகுந்து (logged on), \$ தூண்டுகுறியில் pwd கட்டளையை இவ்வாறு தருகிறார் என வைத்துக் கொள்வோம்:

```
[ilamathi@localhost ilamathi] $ pwd
```

```
/home/ilamathi
```

```
[ilamathi@localhost ilamathi] $
```

எனத் திரையில் காட்டும். /home/ilamathi என்னும் இரண்டாவது வரி முறைமையின் பதிலுரை ஆகும். அதாவது, இளமதியின் இல்லக் கோப்பகத்தின் (home directory) முற்றுநிலைப் பாதைப்பெயரைத் தந்துள்ளது. வேறு தகவல் குறிப்பிட்டுச் சொல்லப்படாதபோது, கட்டளைக்கு அடுத்து வரும் வரியே, அக்கட்டளையின் விடை ஆகும்.

நடப்புக் கோப்பகத்தை மாற்றுதல் (Changing Current Directory)

cd (change directory) என்னும் கட்டளை, நடப்புக் கோப்பகத்திலிருந்து, குறிப்பிட்ட கோப்பகத்துக்கு அழைத்துச் செல்லும். எடுத்துக்காட்டாக, இப்போதைய பயனர் இளமதி அவரது இல்லக் கோப்பகத்திலிருந்து, /usr/bin கோப்பகத்துக்கு மாற விரும்புகிறார். அவர் முதலில், தான் எங்குள்ளோம் என்பதை அறிந்துகொள்ள, நடப்புக் கோப்பகத்தைக் கண்டறிந்து, பிறகு /usr/bin கோப்பகத்துக்கு மாறுவார். கட்டளைத் தூண்டுகுறியில் கீழ்க்காணும் கட்டளைகளைச் செயல்படுத்துவார்:

```
[ilamathi@localhost ilamathi] $ pwd
```

```
/home/ilamathi
```

```
[ilamathi@localhost ilamathi] $ cd /usr/bin
```

```
[ilamathi@localhost bin ] $ pwd
```

```
/usr/bin
```

அடுத்து, நடப்புக் கோப்பகத்தின் தாய்க் கோப்பகத்துக்குச் (parent directory) செல்ல விரும்புகிறார் எனில், \$ தூண்டுகுறியில், cd.. என்னும் கட்டளையைப் பயன்படுத்துவார்.

குறிப்பு: இரட்டைப்புள்ளிகள் (..) தாய்க் கோப்பகத்தின் பாதையைக் குறிக்கின்றன. ஒற்றைப்புள்ளி (.) அதே கோப்பகத்தைக் குறிக்கிறது. cd மற்றும் .. ஆகியவற்றுக்கிடையே ஒரு வெற்று இடவெளி (space) இருக்க வேண்டும். ஆனால் இரண்டு புள்ளிகளுக்கு இடையே வெற்று இடவெளி இருக்கக்கூடாது.

எடுத்துக்காட்டு:

```
[ilamathi@localhost bin] $ cd ..
```

```
[ilamathi@localhost /usr] $ pwd
```

```
/usr
```

```
[ilamathi@localhost /usr] $ cd ..
```

```
[ilamathi@localhost /] $ pwd
```

குமரன், முறைமையில் உள்புகுந்து, தற்போது /usr/bin கோப்பகத்தில் உள்ளார் என வைத்துக்கொள்வோம். அந்தப் பயனர் இப்போது தன்னுடைய இல்லக் கோப்பகம் செல்ல விரும்புகிறார். வெறுமனே cd எனக் கட்டளை தந்தால் போதும். இக்கட்டளை அவரை, அவரது இல்லக் கோப்பகத்துக்கு இட்டுச் செல்லும்.

எடுத்துக்காட்டு:

```
[kumaran@localhost bin] $ cd
```

```
[kumaran@localhost kumaran] $ pwd
```

```
/home/kumaran
```

குறிப்பு: பாதைப் பெயர் எதுவும் தராமல் கொடுக்கப்படும் cd கட்டளை, எப்போதும் ஒரு பயனரை அவரது இல்லக் கோப்பகத்துக்கு இட்டுச் செல்லும்.

குமரன் இப்போது /usr/bin கோப்பகத்தில் இருக்கிறார். அவரது personal கோப்பகத்துக்குச் செல்லத் தீர்மானிக்கிறார். அவரது விருப்பத்தை நிறைவேற்றிக்கொள்ள மிக எளிமையான வழி, /personal என்னும் கோப்பகப் பெயரை நெளிவுக்(~) குறியோடு சேர்த்துத் தரவேண்டும்.

```
[kumaran@localhost bin] $ ~/personal
```

```
[kumaran@localhost personal] $ pwd
```

```
/home/kumaran/personal
```

[kumaran@localhost personal] \$ cd ~

[kumaran@localhost kumaran] \$ pwd

/home/kumaran

குறிப்பு: நெளிவுக் குறி, உங்கள் இல்லக் கோப்பகத்துக்கு மாற்றாகும்.

குமரன் அவருடைய **personal** கோப்பகத்தில் இருக்கிறார், அவருடைய **health** கோப்பகத்துக்குச் செல்ல விரும்புகிறார் எனக் கருதிக் கொள்வோம். **cd..** கட்டளை, **/health** என்பதோடு சேர்த்து இப்பணியை நிறைவேற்றி முடிக்கும். **\$** தூண்டுகுறியில் கீழ்க்காணும் கட்டளையை அவர் தரவேண்டும்.

[kumaran@localhost personal] \$ cd .. /health

[kumaran@localhost health] \$ pwd

/home/kumaran/health

ஒரு கோப்பகத்தின் கீழ் இருக்கும் இன்னொரு கோப்பகம் அக் கோப்பகத்தின் 'உள் -கோப்பகம்' (Sub - Directory) என்று அழைக்கப்படுகிறது.

கோப்பகம் உருவாக்குதல் (Creating a Directory)

குமரன் என்னும் பயனர், அவருடைய **health** கோப்பகத்தின் கீழ் **insurance** என்னும் கோப்பகத்தை உருவாக்க விரும்புகிறார் எனக் கொள்வோம். அவர் ஏற்கெனவே **health** கோப்பகத்தில் உள்ளார். அவர் **mkdir** (**make directory**) என்னும் கட்டளையைப் பயன்படுத்த வேண்டும். **mkdir** என்னும் கட்டளைச் சொல்லுக்கு அடுத்துக் குறிப்பிடுகின்ற பெயரில் ஒரு புதிய கோப்பகம், நடப்புக் கோப்பகத்தின் கீழ் உருவாக்கப்படும். ஆனால் புதிதாக உருவாக்கப்பட்ட கோப்பகம் தானாகவே நடப்புக் கோப்பகமாக மாறாது. புதிதாக உருவாக்கப்பட்ட கோப்பகத்துக்கு மாற விரும்பினால், **cd** கட்டளையைப் பயன்படுத்திக் கொள்ளவேண்டும்.

[kumaran@localhost health] \$ mkdir insurance

இக்கட்டளை **health** என்னும் கோப்பகத்தின் கீழ் **insurance** என்னும் புதிய கோப்பகத்தை உருவாக்கும். புதிய கோப்பகத்தை எங்கே உருவாக்க வேண்டும் என்பதை முழுப்பாதையுடன் **mkdir** கட்டளையில் குறிப்பிட முடியும்.

கோப்பகத்தை நீக்குதல் (Removing a Directory)

ஒரு கோப்பகத்தை உருவாக்க வழியிருக்கிறது எனில், அதனை நீக்கவும் வழியிருக்கவேண்டும். அதற்குரிய கட்டளை என்னவென்று உங்களால் ஊகிக்க முடிகிறதா? ஆம், நீங்கள் நினைப்பது சரியே! `rmdir` (`remove directory`) என்பதுதான் அக்கட்டளை.

எடுத்துக்காட்டு:

```
[kumaran@localhost health] $ rmdir insurance
```

நீங்கள் ஒரு கோப்பகத்தை நீக்க விரும்புகிறீர்கள் எனில், அக் கோப்பகம்,

1. காலியாக (`empty`) இருக்கவேண்டும்.
2. நடப்புக் கோப்பகமாய் (`current directory`) இருக்கக் கூடாது.

குமரன், `insurance` கோப்பகத்தில் எதையும் சேமித்து வைக்கவில்லை. அது காலியாகவே உள்ளது. மேலும், நடப்புக் கோப்பகமாக இருப்பது `health`-தான், `insurance` அல்ல. எனவே, `insurance` கோப்பகத்தை அழிக்க முடிகிறது.

ஏதேனும் சேமித்து வைக்கப்பட்டுள்ள, காலியாக இல்லாத ஒரு கோப்பகத்தை நீக்க விரும்பினால் முதலில் `rm` கட்டளையைப் பயன்படுத்தவேண்டும். அக்கட்டளை பற்றி விரைவில் படிப்போம்.

`mkdir` கட்டளையில் போலவே, `rmdir` கட்டளையிலும் எங்கே உள்ள கோப்பகம் என்பதை உணர்த்த முழுப் பாதையோடு குறிப்பிடமுடியும்.

ஒரு கோப்பகத்தின் உள்ளடக்கத்தைப் பட்டியலிடல்

ஒரு கோப்பகத்தில் உள்ள கோப்புகள் மற்றும் உள்-கோப்புகளின் பெயர்களைப் பட்டியலிட விரும்புகிறீர்கள். உங்கள் விருப்பத்தை நிறைவேற்ற `ls` கட்டளை காத்திருக்கிறது.

எடுத்துக்காட்டாக, பயனர் குமரன், `/home/kumaran` என்னும் கோப்பகத்திலுள்ள கோப்புகள் மற்றும் உள்-கோப்பகங்களின் பெயர்களை அறிய விரும்புகிறார் என வைத்துக்கொள்வோம். அவர் தரவேண்டிய கட்டளை இதுதான்:

```
[Kumaran@localhost kumaran]$ ls /home/kumaran
health  cprogramming  c++programming  personal.
[kumaran@localhost kumaran] $ ls -F
health/  cprogramming  c++ programming  personal/
```


Is கட்டளையில் -F என்னும் தேர்வு, ஒவ்வொரு கோப்பக மற்றும் உள்-கோப்பகப் பெயருக்குப் பின்னாலும் / என்னும் குறியைச் சேர்க்கும்.

குறிப்பு: வெறுமனே Is எனக் கட்டளை தந்தாலும் அதே பட்டியல்தான் கிடைக்கும். காரணம் குமரன், அவருடைய /home கோப்பகத்தில்தான் உள்ளார். (/ என்னும் குறியை உள்-கோப்பகப் பெயர்களின் இறுதியில் சேர்க்காது).

நடப்புக் கோப்பகம் பற்றிய தகவலை அறிய விரும்பினால் வெறும் Is கட்டளை உங்கள் தேவையை நிறைவேற்றும். கோப்புகள் மற்றும் கோப்பகங்கள் பற்றி மேலும் விவரங்கள் தேவையெனில் Is கட்டளையுடன் -l தேர்வைச் சேர்க்க வேண்டும். -l தேர்வு, மறைக்கப்பட்ட கோப்புகள் எவற்றையும் பட்டியலிடாது. ஆனால், Is கட்டளையுடன், -a என்னும் தேர்வு, மறைக்கப்பட்ட கோப்புகள் உட்பட அனைத்துக் கோப்புகள் மற்றும் உள்-கோப்பகங்களைப் பட்டியலிடும். -a மற்றும் -l தேர்வுகளை இணைத்து ஒரே நேரத்தில் பல வழிகளில் தரமுடியும்:

-al, -la, -a -l, -l -a

Is கட்டளையுடன் பயன்படுத்தப்படும் பொதுவான தேர்வுகளைக் கீழே உள்ள அட்டவணையில் காண்க:

தேர்வு	செயல்பாடு
-a	மறைக்கப்பட்ட கோப்புகள் உட்பட அனைத்துக் கோப்புகளையும் பட்டியலிடும்.
-F	கோப்பின் வகையைப் பெயருடன் காட்டும் (கோப்பகத்தைக் கோப்பிலிருந்து பிரித்துக்காட்ட, கோப்பகம் ஒவ்வொன்றின் பெயருக்குப் பின்னாலும் / என்னும் குறியைச் சேர்த்துக் காட்டும்.
-R	பணிபுரியும் கோப்பு மற்றும் அனைத்து உள்-கோப்பகங்களின் உள்ளடக்கத்தையும் பட்டியலிடும்.
-r	கோப்புகள் மற்றும் உள்-கோப்பகங்களின் பட்டியலை தலைகீழ் வரிசையில் காட்டும்.
-s	பட்டியலை கோப்பின் அளவில் (size) வரிசைப்படுத்தும்.
-A	. மற்றும் .. கோப்பகங்கள் தவிர, அனைத்துக் கோப்பகங்களிலும் உள்ள கோப்புகளைக் காட்டும்.

3.6 பிற கட்டளைகள்

திரையைக் கையாளுதல்

`clear` என்னும் கட்டளை திரையைத் துடைத்துவிடும். `tput clear` கட்டளை மூலமாகவும் இதே பணியை நிறைவேற்றலாம்.

```
tput cup 20 20
```

என்னும் கட்டளை, காட்டியை (`cursor`) கிடக்கை(`row`) 20, நெடுக்கை (`column`) 20 -ல் நிறுத்தும். இக்கட்டளை சி-மொழி நிரலாக்கத்திலுள்ள `gotoxy(20,20)` கட்டளைக்கு இணையானது.

உதவியைப் பெறுதல்

ஏதேனும் ஒரு கட்டளைபற்றி, எடுத்துக்காட்டாக `ls` கட்டளை பற்றிய உதவிக் குறிப்புகளைப் பெற விரும்பினால் `man (manual)` கட்டளையைப் பயன்படுத்தவேண்டும். `man` கட்டளை, ஓர் 'எடுபிடி' போலச் செயல்படும். (எல்லா வகையான வேலைகளையும் செய்யக்கூடிய ஒரு பொதுவான வேலையாளர் அல்லது பணியாளரை ஆங்கிலத்தில் `Man Friday` என்றழைப்பர்).

எடுத்துக்காட்டு:

```
$ man ls
```

`man` கட்டளையிடமிருந்து நீங்கள் எதிர்பார்க்கும் உதவி நிலையை (`level of help`) குறிப்பிட முடியும். அந்த நிலைக்குரிய எண்ணை, `man` என்ற கட்டளைச் சொல்லுக்கும், உதவி பெற விரும்பும் கட்டளைச் சொல்லுக்கும் நடுவில் தரவேண்டும். இனி, நிகழ்நிலையில் (`online`) உதவி கிடைக்கும்.

echo கட்டளை

`echo` கட்டளை, சி-மொழி நிரலாக்கத்தில் பயன்படுத்தப்படும் `printf()` மற்றும் `\n` ஆகியவை இரண்டையும் இணைத்தது போலச் செயல்படும். பயனருக்கு ஒரு செய்தியைத் திரையில் காட்ட விரும்பினால், `echo` கட்டளையைப் பயன்படுத்தலாம்.

எடுத்துக்காட்டு:

```
$echo "Please enter your name"
```

இக்கட்டளை, `Please enter your name` என்கிற செய்தியைத் திரையில் காட்டும். அடுத்த வரியில், காட்டி (`cursor`) காத்திருக்கும். அதே

வரியில் செய்தியை அடுத்துக் காட்டியை நிறுத்த, echo கட்டளையுடன் சேர்த்து, -n தேர்வைப் பயன்படுத்தவேண்டும்.

\$echo -n "Please enter your name"

இரட்டை மேற்கோள் குறிகள் (double quotes - "") படிப்பதற்கு எளிமை சேர்க்கின்றன. (improve read ability). மற்றபடி, மேலே உள்ள கட்டளையை இவ்வாறு அமைத்தாலே போதும்.

\$echo -n Please enter your name

தொகுப்புரை

- * லினக்ஸ் முறைமையில் உள்புகுதல் (logging on) மூலமாக, ஒரு லினக்ஸ் அமர்வுக்குள் நுழைய முடியும். logout அல்லது exit கட்டளை மூலமாக, லினக்ஸ் அமர்விலிருந்து வெளியேறலாம்.
- * உங்களுக்குரிய கடவுச்சொல்லை மாற்றி அமைக்கலாம்.
- * கோப்புகளை வேகமாக அணுக வகைசெய்யும் படிமரபுக் கோப்பு முறைமையை (Hierarchical File System) லினக்ஸ் வழங்குகிறது.
- * லினக்ஸில் சாதாரணக் கோப்புகள் (Ordinary Files), கோப்பகக் கோப்புகள் (Directory Files), சிறப்புக் கோப்புகள் (Special Files) எனப் பல்வேறு வகைக் கோப்புகள் உள்ளன.
- * கோப்புகளைப் பயன்படுத்தும் பயனர்கள் பல வகைப்படுவர்:
 - ❖ மூலப் பயனர் (Root user)
 - ❖ கோப்பு உரிமையாளர் (File Owner)
 - ❖ குழு உரிமையாளர் (Group Owner)
 - ❖ பிற பயனர்கள் (Other users)
- * ஒரு கட்டளையின் பொதுவான வடிவம்:

Command	Option	Argument
---------	--------	----------
- * அடிக்கடி பயன்படுத்தப்படும் சில கட்டளைகள்:
 - ❖ pwd - தற்போது பணிபுரியும் கோப்பகத்தைக் காட்டும்
 - ❖ man - உதவிக் குறிப்புகளை வழங்கும்
 - ❖ mkfs - நெகிழ்வட்டை வடிவமைக்கும்
 - ❖ cd - தற்போது பணிபுரியும் கோப்பகத்திலிருந்து மாறும்
 - ❖ mkdir - ஒரு புதிய கோப்பகத்தை உருவாக்கும்

- ❖ rmdir ஒரு காலியான கோப்பகத்தை நீக்கும்
- ❖ ls -l ஒரு கோப்பகத்தின் உள்ளடக்கத்தைக் காட்டும்
- ❖ echo ஒரு செய்தியைத் திரையில் காட்டும்

பயிற்சி வினாக்கள்

I. புள்ளியிட்ட இடங்களை நிரப்புக

1. _____ இயக்க முறைமையை உருவாக்கியவர் கென் தாம்ப்சன்.
2. மினிக்ஸ் இயக்க முறைமையை உருவாக்கியவர் _____.
3. மூலப் பயனர் (எஸ்ஏ) எந்தப் பயனரின் _____ யும் மாற்று வதற்கு அதிகாரம் படைத்தவர் ஆவார்.
4. ஒரு பயனர் அவருடைய _____ பெயரையும், _____ யும் தந்து முறைமையின் உள்ளே நுழையலாம்.
5. ஒரு பயனர் தன்னுடைய பழைய (இப்போதைய) கடவுச்சொல் லுடன் _____ யும் தந்து, அவருடைய கடவுச்சொல்லை மாற்றியமைக்கலாம்.
6. லினக்ஸ் _____ பயனர் முறைமை ஆகும்.
7. மூலப் பயனருக்கான தூண்டுகுறி _____ ஆகும்.
8. நடப்புக் கோப்பகத்தைக் கண்டறிய _____ கட்டளை பயன் படுகிறது.
9. cd மற்றும் .. இரண்டுக்கும் இடையே குறைந்தது ஒரு _____ இருக்கவேண்டும்
10. _____ என்னும் குறி, உங்கள் இல்லக் கோப்புறையின் முழுப் பாதையை உருவாக்கிறது.

II. சரி அல்லது தவறு எனக் குறிப்பிடுக

1. கென் தாம்ப்சன் மினிக்ஸை உருவாக்கினார்.
2. ஹங்கேரி நாட்டைச் சேர்ந்த ஒரு சிலரே லினக்ஸை மேம்படுத்தினர்.
3. லினக்ஸ் அதன் எளிய வடிவில் 4 எம்பி நினைவகத்தையே எடுத்துக்கொள்ளும்.
4. n.x.y என்னும் பதிப்பெண்ணில் x ஒற்றைப்படை எனில் அப்பதிப்பு நிலைத்த (stable) பதிப்பாகும்.

5. பயனர், கடவுச்சொல்லை மட்டும் பயன்படுத்தி உள்புக முடியும்.
6. logout மற்றும் exit ஆகிய இரு கட்டளைகளையும் சேர்த்துப் பயன்படுத்தி வெளியேற முடியும்.
7. பயனர் மட்டுமே அவருடைய கடவுச்சொல்லை மாற்ற முடியும்.
8. தேடலை எளிமைப்படுத்த, லினக்ஸில் படிமுறைக் கட்டமைப்பு பயன்படுத்தப்படுகிறது.
9. எந்த விதிவிலக்கும் இன்றி, அனைத்துப் பயனர்களும் \$ தூண்டு குறியையே பெறுகின்றனர்.
10. முறைமை நிர்வாகிக்கு (System Administrator) 'மீப் பயனர்'(Super user) என்ற பெயரும் உண்டு.
11. ஒரு கோப்பகத்திலிருந்து வேறொரு கோப்பகத்துக்கு மாற, change directory என்னும் கட்டளையைப் பயன்படுத்தவேண்டும்.

III. விடைதருக

1. மீப் பயனர் (Super user) என்பவர் யார்?
2. உங்களின் தற்போதைய கடவுச்சொல்லை எவ்வாறு மாற்றியமைப்பீர்கள்?
3. உங்கள் கடவுச்சொல்லை மாற்றியமைக்கும்போது பின்பற்ற வேண்டிய விதிமுறைகள் யாவை?
4. நீங்கள் தற்போது பணிபுரியும் கோப்பகத்தை எவ்வாறு அறிந்து கொள்வீர்கள்?
5. மூலப் பயனரின் சலுகை உரிமைகள் யாவை?
6. மறைக்கப்பட்ட ஒரு கோப்பின் பெயரை எவ்வாறு அறியலாம்?
7. உங்கள் கோப்புகளை அளவின் அடிப்படையில் எவ்வாறு வரிசைப்படுத்திக் காட்டுவீர்கள்?
8. rmdir கட்டளையைப் பயன்படுத்தி, ஒரு கோப்பகத்தை நீக்குவதற்குத் தேவையான நிபந்தனைகள் எவை?
9. நெகிழ்வட்டில் உள்ள ஒரு கோப்பினை முறைமை நிர்வாகியின் உதவியுடன் எவ்வாறு இயக்குவீர்கள்?

10. ஒரு புதிய கோப்பகத்தை எவ்வாறு உருவாக்குவீர்கள்?
11. கோப்புகள், கோப்பகங்கள், உள்-கோப்பகங்கள் ஆகியவற்றை எவ்வாறு பட்டியலிடுவீர்கள்?
12. man கட்டளையின் செயல்பாடு என்ன?

3.7 கோப்புக் கட்டளைகள்

கோப்பின் உள்ளடக்கத்தைக் காட்டுதல்

cat கட்டளை கோப்பின் உள்ளடக்கத்தை வெளிக்கொணரும். ஆனால் more கட்டளை இன்னும் நிறையப் பலன் தரும். cat கட்டளை, குறிப்பிட்ட கோப்பின் உள்ளடக்கத்தைத் திரையில் காட்டும். கோப்பு பெரியது எனில், உள்ளடக்கம் திரையில் வேகமாக ஓடி நிற்கும். முடிவில் நீங்கள் காண்பது கடைசிப் பக்கத்தை மட்டுமே. அத்தகைய சூழ்நிலைகளில் more கட்டளை கைகொடுக்கும். அது, ஒரு நேரத்தில் ஒரு பக்கத்தை மட்டுமே காட்டும். அடுத்த பக்கத்துக்கு அல்லது முந்தைய பக்கத்துக்கு நகர, முறையே f (forward) அல்லது b (backward) விசைகளை அழுத்தவேண்டும். கறாராகச் சொல்வதெனில், cat, more ஆகியவை வடிகட்டிகள்(filters) ஆகும். அவை தம்வழியே செல்லும் தரவுகளை வடிகட்டி அனுப்பு கின்றன.

tee என்னுமொரு கட்டளை உண்டு. cat கட்டளையின் பணியைப் போல் இருவேறு பணிகளைச் செய்து முடிக்கும். tee கட்டளை ஒரு T குழாய்போலச் செயல்படும். இந்தக் குழாய் இரண்டு குழல்களால் ஆனது. முதல்பகுதி ஒரு கிடைமட்டக் குழல். இரண்டாம் பகுதி ஒரு செங்குத்துக் குழல். T குழாய் (கிடைமட்டத் தளத்தில் கிடத்தப்பட்டது) வழியாக நீரைச் செலுத்தினால் ஒரே நேரத்தில் நீர் கிடைமட்டமாகவும் செங்குத்தாகவும் பாயும். அதேபோன்று tee கட்டளை அடிப்படை உள்ளீட்டகத்திலிருந்து (Standard input) உள்ளீட்டைப் பெற்று, அந்த விவரங்களைத் திரையில் காட்டும் (cat கட்டளை போல). அதே விவரங்களை, tee கட்டளையை அடுத்துக் குறிப்பிடப்படும் ஒரு கோப்பிலும் சேமிக்கும். ஆக, tee கட்டளை ஒரே நேரத்தில் இரண்டு வேலைகளைச் செய்து முடிக்கிறது.

லினக்ஸில் அனைத்துக் கோப்புகளும் தொடர்ச்சியான பைட்டுகளின் தாரையாக (stream) ஒழுங்கமைக்கப்பட்டுள்ளன. லினக்ஸில், 'பைட்-தாரைக் கோப்பு' (byte-stream file) என்று சொல்லப்படும் ஒரேயொரு அடிப்படை வகைக் கோப்பு மட்டுமே உண்டு. உள்ளீட்டுத் தரவுத் தாரை, 'அடிப்படை உள்ளீடு' (standard input) என்றும், வெளியீட்டுத் தாரை, 'அடிப்

படை வெளியீடு' (standard output) என்றும் அழைக்கப்படும். உள்ளீடு செய்யப்படும் தரவுகள், தொடர்ச்சியான பைட்டுத் தொகுதியால் ஆன தரவுத் தாரையாக (data stream) மாற்றப்படுகின்றன. இதுதான் 'அடிப்படை உள்ளீடு' என அழைக்கப்படுகிறது. பொதுவாக அடிப்படை உள்ளீடு என்பது விசைப்பலகையோடு (keyboard) இணைக்கப்பட்டிருக்கும். அடிப்படை வெளியீடு என்பதும் தொடர்ச்சியான பைட்டுத் தொகுதியால் ஆன தரவுத் தாரையே ஆகும். பொதுவாக, 'அடிப்படை வெளியீடு' என்பது அச்சுப்பொறியுடன் இணைக்கப்பட்டிருக்கும். அடிப்படை உள்ளீட்டை நெகிழ்வட்டு போன்றவற்றுக்குத் திசைதிருப்ப முடியும். அதே பாணியில், அடிப்படை வெளியீட்டையும் குறுவட்டு, நெகிழ்வட்டு போன்ற ஒரு சேமிப்புச் சாதனத்துக்குத் திசைதிருப்ப முடியும்.

மேற்கண்ட கட்டளைகளுக்குரிய எடுத்துக்காட்டுகளைக் காண இருக்கிறீர்கள். இனிமேல், தூண்டுகுறியாக [ilamathi@localhost ilamathi]\$ எனத் தருவதற்குப் பதிலாக வெறுமனே \$ குறி மட்டுமே கட்டளை வரியில் தரப்படும்.

file1 என்னும் கோப்பின் உள்ளடக்கத்தைத் திரையில் காண விரும்புகிறீர்கள். கீழேயுள்ள எடுத்துக்காட்டில் விளக்கப்பட்டுள்ளதுபோலக் கட்டளை தரவேண்டும். file1- ன் உள்ளடக்கம்,

It is a fun.

You are encouraged to work with the Linux

என்று இருப்பதாகக் கொள்வோம்.

\$cat file1

என்ற கட்டளையின் வெளியீடு,

It is a fun.

You are encouraged to work with the Linux.

என்று கிடைக்கும்.

file1 - ன் உள்ளடக்கத்தைத் திரையில், பக்கம் பக்கமாகப் பார்வையிட விரும்புகிறீர்கள் எனில், கட்டளையை இவ்வாறு அமைக்க வேண்டும்:

\$ more file1

இதன் வெளியீடு,

It is a Fun

You are encouraged to work with the Linux.

அதாவது, **cat** கட்டளைக்கும் **more** கட்டளைக்கும் வேறுபாடு எதுவும் காண முடியவில்லை. ஆனால், **file1**-ன் உள்ளடக்கம் ஒரு பக்கத்துக்கும் அதிகமாக இருப்பின், **cat** கட்டளையில் கடைசிப் பக்கத்தை மட்டுமே பார்ப்பீர்கள். **more** கட்டளையில் பக்கம் பக்கமாக நிறுத்திப் பார்க்கமுடியும்.

திசைதிருப்பல் (**redirection**), குழாய்ச் செலுத்தல் (**pipng**) பற்றிக் கற்றுக்கொண்ட பிறகு **cat**, **more**, **tee** ஆகிய கட்டளைகளுக்கு நிறைய எடுத்துக்காட்டுகளைப் பார்ப்போம்.

மழைநீர் சேமிப்பு அமைப்பு இல்லையெனில், வீடுகளின் மொட்டை மாடிகளில் திரளுகின்ற மழைநீர் பெரும்பாலும் தெருக்களில் விடப்படுகின்றன. அதாவது, மொட்டை மாடி நேடியாகத் தெருவோடு இணைக்கப்பட்டுள்ளது. மழைநீர் சேகரிப்புத் திட்டம் அறிமுகப்படுத்தப்பட்டபின், மழை பெய்யும்போது திரளுகின்ற நீரை மக்கள், கிணற்றுக்கும், சேகரிப்புக் கிடங்குகளுக்கும் திசைதிருப்பிவிடுகின்றனர். லினக்ஸ் முறைமையும், வெளியீடு அல்லது உள்ளீடுகளை திரை அல்லது விசைப்பலகைக்கு அல்லாமல் கோப்புகளுக்குத் திசைதிருப்ப முடியும். (லினக்ஸைப் பொறுத்தவரை, அடிப்படை உள்ளீடு, அடிப்படை வெளியீடு, திரையகம், விசைப் பலகை அனைத்தையும் கோப்புகளாகவே கருதிக் கொள்கிறது). திசை திருப்ப செயற்குறி (>) - "விடப்பெரிது" குறியீடு - வெளியீட்டைத் திசைதிருப்பும் பணியைச் செய்கிறது. மற்றொரு திசை திருப்பு செயற்குறி (<) - "விடச்சிறிது" குறியீடு - உள்ளீட்டைத் திசைதிருப்பும் பணியைச் செய்கிறது. வெளியீட்டுத் திசைதிருப்பு செயற்குறி, இடப்பக்கம் குறிப்பிடப்படும் கோப்பின் (> அடையாளத்துக்கு முன்னதாகத் தரப்படும் கோப்பின்) உள்ளடக்கத்தை, வலப்பக்கம் தரப்படும் கோப்புக்கு (> அடையாளத்துக்குப் பிறகு தரப்படும் கோப்புக்கு) திசைதிருப்பி விடுகிறது.

\$cat file1

என்ற கட்டளை **file1**-ன் உள்ளடக்கத்தைத் திரையில் காட்டுகிறது எனப் பார்த்தோம். இந்த வெளியீட்டை **file2** என்னும் கோப்புக்கு அனுப்ப (திசை திருப்ப) விரும்புகிறீர்கள் எனில் கீழே காணும் கட்டளையைத் தர வேண்டும்:

\$cat file1 > file2

\$cat file2

என்று கட்டளை தந்து, **file2** - ன் உள்ளடக்கத்தைத் திரையில் காணுங்கள். அது, **file1**- ன் அப்பட்டமான நகலாக இருக்கும். மேற்கண்ட கட்டளையின் வெளியீடு அடிப்படை வெளியீட்டுக்கு அனுப்பப்படுகிறது. திசை திருப்பல் எதுவும் இல்லை என்பதால், அடிப்படை வெளியீடு திரைக்குத் திருப்பிவிடப்படுகிறது. ஆனால் **\$cat file1 > file2** கட்டளையில், **cat file1** என்னும் பகுதி, வெளியீட்டை அடிப்படை வெளியீட்டுக்கு அதாவது திரைக்கு அனுப்பி வைக்கிறது. ஆனால், அடுத்துள்ள திசைதிருப்பு செயற்குறி (>) திரைக்கு அனுப்பப்படும் வெளியீட்டைத் தடுத்து நிறுத்தி, அவ்வெளியீட்டை **file2**- க்கு அனுப்பிவைக்கிறது. (திசைதிருப்புகிறது). **file2** ஏற்கனவே இல்லையெனில், புதிதாக உருவாக்கப்படும். **file1**-ன் உள்ளடக்கம், **file2**-ல் நகலெடுக்கப்படும். **file2** என்னும் கோப்பு ஏற்கெனவே இருந்தால் என்ன ஆகும்? அதிலுள்ள விவரங்கள் அழிக்கப்பட்டு, **file1**-ன் விவரங்கள் அதில் எழுதப்படும்.

இக்கட்டளையில் நாம் கண்ணில் காண்பதைவிட மேலும் அதிகமான பின்னணி விவரங்கள் உள்ளன. கட்டளையில் **cat file1** என்ற பகுதியே > குறிக்கு முதலில் இருக்கிறது. ஆனாலும் **file2** பகுதியே முதலில் செயல்படுத்தப்படும். **file2** ஏற்கெனவே இருக்கும் எனில் அதிலுள்ள விவரங்கள் அழிக்கப்படும். இடப்பக்கம் உள்ள **file1** கோப்பின் உள்ளடக்கம், அடிப்படை வெளியீட்டுக்கு அனுப்பி வைக்கப்படும். இந்த வெளியீட்டே **file2**-க்கு உள்ளீடாக அமைகிறது. சுருங்கச்சொல்லின், **file1**-ன் உள்ளடக்கம் **file2**-ல் எழுதப்படுகிறது. மேற்கண்ட விளக்கத்திலிருந்து, கீழே காணும் கட்டளை செயல்படாது என்பதை நீங்கள் அறியலாம்:

\$cat file1 > file1

மேற்கண்ட கட்டளையைச் செயல்படுத்தினால், வலப்பக்கப் பகுதியே முதலில் செயல்படும். **file1** ஏற்கெனவே இருப்பதால் அதனுடைய உள்ளடக்கம் அழிக்கப்பட்டுவிடும். இனி, இடப்பக்கப் பகுதி செயல்படும். **file1** கோப்பில் இப்போது எதுவும் இருக்காது. எனவே கட்டளை செயல்படாது.

இவ்வாறு திசைதிருப்பல் செயல்பாடு காரணமாக, ஏற்கெனவே இருக்கும் கோப்பின் உள்ளடக்கம் அழிக்கப்பட்டுப் புதிய தகவல் எழுதப்படுவதைத் தடுப்பதற்கு **noclobber** என்னும் வசதியைப் பயன்படுத்திக்கொள்ளலாம். இது, இருக்கும் கோப்பில் மேலெழுதப்படுவதைத் (**overwriting**) தடுக்கிறது. இதைத் தவிர வேறொரு வழியும் உள்ளது. ஏற்கெனவே இருக்கும் கோப்பில் மேலெழுதப்படுவதைத் தவிர்க்கப் பின்சேர்ப்புச்

(append) செயற்குறியைப் (>>) பயன்படுத்தலாம். பின்சேர்ப்புச் செயற்குறி, இடப்பக்கம் உள்ள கோப்பின் உள்ளடக்கத்தை, வலப்பக்கம் உள்ள கோப்பில், ஏற்கெனவே உள்ள விவரங்களின் கீழாகச் சேர்த்துவிடும்.

எடுத்துக்காட்டாக,

\$cat file1 >> file2

என்னும் கட்டளை file1- ன் உள்ளடக்கத்தை, file2 - ல் ஏற்கெனவே உள்ள உள்ளடக்கத்தின் கீழாகச் சேர்த்துவிடும்.

அடிப்படை உள்ளீடு (The Standard Input)

பல லினக்ஸ் கட்டளைகள், அடிப்படை உள்ளீடு வழியாகவே தரவுகளைப் பெறுகின்றன. அடிப்படை உள்ளீடு என்பது ஒரு சாதனத்துடன் அல்லது ஒரு கோப்புடன் இணைக்கப்பட்டிருக்கும். முன்னியல்பாக (by default) அடிப்படை உள்ளீடு, விசைப்பலகையுடன் இணைக்கப்பட்டுள்ளது. விசைப்பலகை மூலம் பதிவு செய்யப்படும் எழுத்துகள், அடிப்படை உள்ளீட்டுக்கு எடுத்துச் செல்லப்படும், அதன்பின் கட்டளைக்கு அனுப்பி வைக்கப்பட்டு, அதற்கேற்ப செயல்படுத்தப்படும்.

cat கட்டளை எவ்வித செயலுருபும் இல்லாவிடில், அடிப்படை உள்ளீட்டிலிருந்து, உள்ளீடுகளை ஏற்றுக்கொள்ளும். cat கட்டளைக்குரிய தகவலை விசைப்பலகையின் மூலமாக உள்ளிடவேண்டும். அவை அடிப்படை உள்ளீடு வழியாக cat கட்டளைக்கு எடுத்துச்செல்லப்படும்.

எடுத்துக்காட்டு:

\$cat

இக்கட்டளையைத் தந்தவுடன், காட்டி (cursor) அடுத்தவரிக்கு வந்து காத்திருக்கும்.

This command expects data from the standard input

(விசைப்பலகை மூலம் உள்ளீடு)

This command expects data from the standard input (வெளியீடு)

Now you have to enter the data from the keyboard

(விசைப்பலகை மூலம் உள்ளீடு)

Now you have to enter the data from the keyboard (வெளியீடு)

Ctrl+D

\$

cat கட்டளையைத் தந்தபின், விசைப்பலகையிலிருந்து ஒரு வரியை உள்ளீடாகத் தருகிறீர்கள். முதல்வரி முடிந்ததும், அச்செய்தி இடையகத் திலிருந்து (**buffer**) அடிப்படை உள்ளீட்டுக்கு எடுத்துச்செல்லப்படுகிறது. அங்கிருந்து **cat** கட்டளைக்கு உள்ளீடாகத் தரப்படுகிறது. அங்கே திசை திருப்பு செயற்குறி இல்லை என்பதால் அச்செய்தியை **cat** கட்டளை திரையில் காட்டுகிறது. மேலே இரண்டாவதாக இடம்பெற்றுள்ள வரி, **cat** கட்டளை மூலம், அடிப்படை உள்ளீடு வழியாகத் திரைக்கு அனுப்பப் பட்டதாகும். அதேபோல, மூன்றாவது வரி, பயனர் விசைப்பலகை மூலமாக உள்ளீடு தந்ததாகும். நான்காவது வரி, முறைமையின் பதிலுரையாகும். உங்கள் பணியை முடித்தபிறகு, நீங்கள் பணியை முடித்துவிட்டீர்கள் என்பதைக் கணிப்பொறிக்கு உணர்த்தவேண்டும். கடைசி வரியில் **Ctrl+D** எனத் தரும் கட்டளை, இதை உணர்த்திவிடுகிறது.

குறிப்பு: தரவுகளை (**data**) ஓர் அணைக்கட்டில் தேங்கி நிற்கும் நீருக்கு ஒப்பிடலாம். ஏராளமான நீர்வரத்து இருக்கும்போது அதைப் பாசனத்துக்கு அனுப்பும் முன்பாக, அணையின் தேக்கி வைக்கிறோம். அணையானது, நீரைச் சேமித்து வைத்து, பாசனத்துக்குத் தேவை ஏற்படும்போது, அல்லது அணை நிரம்பும்போது, சீரான முறையில் நீரை அனுப்பி வைக்கிறது. இடையகத்தை அணைக்கு ஒப்பிடலாம். உள்ளீடு செய்யப்படும் தரவுகள் இடையகத்தில் தேக்கிவைக்கப்பட்டு, அடிப்படை உள்ளீட்டுக்குச் சீரான முறையில் அனுப்பி வைக்கப்படுகின்றன. லினக்ஸில் **Ctrl+D** குறியீடு, கோப்பு முடிவுக் குறியுரு (**end-of-file character**) ஆகும்.

cat கட்டளையை வெளியீட்டுக்கான திசைதிருப்பு செயற்குறி (>) யுடன் சேர்த்துத் தரமுடியும்.

எடுத்துக்காட்டு:

\$ cat > file3

The typed in material will be redirected

to the cat command through the std input (விசைப்பலகை மூலம்

and this message is sent to the file after ">"

உள்ளீடு)

symbol.

Ctrl+D

\$ cat file3

**The typed in material will be redirected
to the cat command through the std input
and this message is sent to the file after ">"
symbol. \$**

உள்ளீட்டுத் திசைதிருப்பு செயற்குறி (<)
(Input Redirection Operator)

தரவுகள் பொதுவாக, விசைப்பலகை வழியாக அடிப்படை உள்ளீட்டுக்கு அனுப்பிவைக்கப்படுகின்றன. அடிப்படை உள்ளீட்டை, கோப்புகளிலிருந்து தரவுகளைப் பெறவும் பயன்படுத்திக்கொள்ள முடியும். உள்ளீட்டுத் திசைதிருப்பு செயற்குறி (<) மூலம் இது சாத்தியமாகிறது. cat கட்டளை, file3 என்னும் கோப்பிலிருந்து (விசைப்பலகையிலிருந்து அல்ல) தரவுகளைப் பெறுவதற்குக் கீழே காணும் கட்டளையைத் தர வேண்டும்.

\$cat < file 3

**The typed in material will be redirected
to the cat command through the std input
and this message is sent to the file after ">"
symbol.**

வெளியீட்டுத் திசைதிருப்பு செயற்குறி (>) இல்லாத காரணத்தால், செய்தி திரைக்கு அனுப்பப்படுகிறது. உள்ளீட்டுத் திசைதிருப்பு செயற்குறி (<) file3-ன் உள்ளடக்கத்தை அடிப்படை உள்ளீட்டுக்கு அனுப்பிவைக்கிறது. cat கட்டளை அடிப்படை உள்ளீட்டைப் படித்து, file3-ன் உள்ளடக்கத்தைத் திரையில் காட்டுகிறது. file3-ன் தரவுகளை, அடிப்படை உள்ளீட்டுக்குத் திசைதிருப்பி file4-ல் எழுதப்படுமாறு, அடிப்படை வெளியீட்டையும் திசைதிருப்ப வேண்டுமெனில், கீழே உள்ள கட்டளையைத் தரவேண்டும்:

\$cat < file3 > file4

குழாய்கள் (Pipes)

ஒரு கட்டளையின் வெளியீட்டை இன்னொரு கட்டளைக்கு உள்ளீடாகத் தரவேண்டுமெனில் திசைதிருப்பு செயற்குறி பயன்தராது. திசைதிருப்பு செயற்குறி கோப்புகளின்மீது மட்டுமே செயல்படும். இந்தச் சூழ்நிலைகளில் குழாய் (Pipe) கைகொடுக்கிறது.

கோப்புகளுக்கும் கட்டளைகளுக்கும் இடையேயுள்ள வேறுபாட்டை நீங்கள் தெளிவாகப் புரிந்துகொள்ள வேண்டும். ஓர் எடுத்துக்

காட்டைப் பார்ப்போம். ஒரு கிடங்கில் மரக்கட்டைகளைச் சேமித்து வைத்துள்ளீர்கள். மரக்கட்டையிலிருந்து நாற்காலிகள் செய்வதற்கு ஒரு தச்சரையும், நாற்காலிகளுக்குச் சாயம் தீட்ட ஒரு சாயம் பூசுநரையும் பணியமர்த்தியுள்ளீர்கள். இப்போது மரக்கட்டையிலிருந்து செய்யப்பட்ட நாற்காலிகள், கொண்டுசெல் பட்டை (Conveyor Belt) மூலமாக, சாயம் பூசுநரின் இடத்துக்கு அனுப்பிவைக்கப்படுகின்றன. அதாவது, நாற்காலிகள் தச்சரிடமிருந்து, சாயம்பூசுநருக்கு எடுத்துச் செல்லப்படுகின்றன. கொண்டு செல் பட்டை குழாய் போலப் பயன்படுகிறது. அதாவது, பண்டங்கள், ஒரு பணியாளரிடமிருந்து இன்னொரு பணியாளருக்குக் குழாய் (கொண்டுசெல் பட்டை) மூலமாக அனுப்பிவைக்கப்படுகின்றன. லினக்ஸில் குழாய் செலுத்தல் செயற்பாடு (Piping Operation) இது போன்றதுதான். தரவுகள், ஒரு கட்டளையிலிருந்து இன்னொரு கட்டளைக்குக் குழாய் வழியாக அனுப்பிவைக்கப்படுகின்றன.

நாற்காலிகள் சாயம் பூசப்பட்டபின், லாரி மூலமாகக் கிடங்குக்கு அனுப்பி வைக்கப்படுகின்றன. இம்முறை, பணியாளர் இடத்திலிருந்து, கிடங்குக்கு நாற்காலிகளை எடுத்துச் செல்ல, லாரி பயன்படுத்தப்படுகிறது. இது திசைதிருப்பல் ஆகும்.

கோப்பு என்பது ஒரு விந்தையான கிட்டங்கி ஆகும். அதிலுள்ள பொருள்களை வேறு இடத்துக்கு அனுப்பிய பின்னும் பொருள்கள் தீர்ந்து போவதில்லை. அனுப்பிவைக்கப்பட்ட பொருள்களின் நகல் அங்கேயே அப்படியே இருக்கும்.

கோப்பு என்பது தரவுகளைச் சேமித்து வைப்பதற்கான ஒரு சேமிப்பு ஊடகம் (மேலேயுள்ள எடுத்துக்காட்டில் கிட்டங்கி போல). கட்டளை என்பது, ஒரு ஆணைத் தொகுதியைச் செயல்படுத்துகின்ற நிரல் (எடுத்துக்காட்டில் பணியாட்களைப் போல). ஒரு கோப்பினில் தரவுகளைச் சேமித்து வைக்கமுடியும்; கோப்பிலிருந்து அவற்றை மீட்டெடுக்கவும் முடியும். ஒரு கட்டளை மூலமாகக் கோப்பிலிருந்து தரவுகளைப் படிக்கமுடியும், கோப்பில் தரவுகளைச் சேமிக்க முடியும்; ஆனால் கட்டளை தானே தரவுகளைச் சேர்த்து வைக்க முடியாது. திசைதிருப்பல், வெளியீட்டை ஒரு கோப்பினில் எழுதுகிறது. ஆனால் குழாய்கள், வெளியீட்டை வேறொரு கட்டளைக்கு அனுப்புகின்றன.

ஒரு கோப்பின் உள்ளடக்கத்தை அச்சிட விரும்புகிறீர்கள். cat கட்டளை ஒரு கோப்பின் உள்ளடக்கத்தைப் பெற்று அதனை அடிப்படையிலே வெளியீட்டுக்கு அனுப்புகிறது என்பதை அறிவீர்கள். அதுபோல, cat கட்டளையின் வெளியீட்டை lpr (line print) என்னும் இன்னொரு கட்ட

ளைக்குக் குழாய்வழி செலுத்தலாம். lpr கட்டளை, அடிப்படை வெளியீட்டை உள்ளீடாக ஏற்று அதனை அச்சப்பொறிக்கு அனுப்புகிறது. இங்கே, இரண்டு கட்டளைகளுக்கு இடையே தரவுகள் பரிமாறிக் கொள்ளப்படுகின்றன. இந்தப் பணிக்குத் திசைதிருப்பல் ஏற்றதல்ல என்பதை அறிவீர்கள். எனவே குழாய்களின் உதவியைத்தான் நாடவேண்டும்.

ஒரு குழாயானது அக்குழாய்க்கு முன்னதாக உள்ள கட்டளையிலிருந்து தரவுகளைப் பெற்று, குழாய்க்கு அடுத்துள்ள கட்டளைக்கு உள்ளீடாகத் தருகிறது. குழாய் செலுத்தும் குறியீடு, செங்குத்து கோடு “|” ஆகும். மேற்கண்ட பணிக்கான கட்டளையைப் பாருங்கள்:

\$cat file3 | lpr

file3-ன் உள்ளடக்கம், தற்போது லினக்ஸ் முறைமையில் இணைக்கப்பட்டுள்ள வரி அச்சப்பொறிக்கு (line printer) அனுப்பிவைக்கப்படுகிறது.

விசைப்பலகை வழியாகத் திரைக்கு விவரங்களை அனுப்பிவைப்பது எப்படி என்பதை ஏற்கெனவே அறிவீர்கள். எந்தக் கோப்புப் பெயரும் தரப்படாத வெறும் cat கட்டளை அந்தப் பணியைச் செய்யும். வெளியீட்டைத் திரையில் காட்டுவதற்குப் பதிலாக, வரி அச்சப் பொறிக்கு அனுப்பிவைக்க என்ன செய்வீர்கள்? ஆகா! விடையைக் கண்டுபிடித்து விட்டீர்கள்! cat மற்றும் lpr கட்டளைகள் இரண்டும் சேர்ந்து உங்கள் விருப்பத்தை நிறைவேற்றி வைக்கும் அல்லவா? இவையிரண்டும் கட்டளைகள் என்பதால் அவற்றை “|” குறி மூலமாக இணைக்க வேண்டும் என்பதை நினைவில் கொள்க. முழுக் கட்டளையையும் ஊகித்து விட்டீர்களா? ஆம், உங்கள் ஊகம் சரியே! இதோ விடை:

\$cat | lpr

The printer should be made ready.

Interesting things are going to be printed.

Ctrl+D

மேற்கண்ட செய்தி (Ctrl+D தவிர்ந்து) அப்படியே அச்சப்பொறியில் அச்சிடப்படும்.

ஒரு கோப்பிலுள்ள (file3 என்க) விவரங்களை வரிவரியாக வரி எண் இட்டு அச்சிடவேண்டிய தேவை ஏற்படலாம். கீழே உள்ள கட்டளை இப்பணியை நிறைவேற்றும்:

\$ cat -n file3 | lpr

`cat` கட்டளை `-n` தேர்வுடன் சேர்ந்து, `file3` -ன் உள்ளடக்கத்தை, ஒவ்வொரு வரியாக வரியெண் இட்டு, அடிப்படை வெளியீட்டுக்கு அனுப்புகிறது. `cat file3`-ன் உள்ளடக்கத்தை, வரியெண்ணிட்டு முன்னியல்பாயுள்ள அச்சுப்பொறிக்கு அனுப்பி வைக்குமாறு `cat` கட்டளையை `-n` தேர்வு வலியுறுத்துகிறது. இப்போது, உங்கள் பழைய நண்பரான `more`- ஐச் சந்திக்கப் போகிறீர்கள். பெரிய கோப்பின் உள்ளடக்கத்தைத் திரையில் காட்ட `cat` கட்டளையைப் பயன்படுத்தினால், நீங்கள் கடைசிப் பக்கத்தை மட்டுமே திரையில் காணமுடியும் என்பதை ஏற்கெனவே அறிவீர்கள். இந்தச் சிக்கலைத் தவிர்க்க, `more` கட்டளையைக் கீழே கண்டவாறு பயன்படுத்தலாம்:

\$cat file3 | more

`file3`- ன் உள்ளடக்கத்தை வரி எண்களோடு திரையில் காட்ட வேண்டிய தேவை ஏற்படுகிறது. மேலே உள்ள கட்டளையைக் கீழே உள்ளவாறு மாற்றியமைக்கவும்:

\$cat -n file3 | more

இந்தக் கட்டளையில் ஒன்றுக்கு மேற்பட்ட கோப்புகளைக் குறிப்பிட முடியும்.

\$cat -n file1 file2 file3 | more

குறிப்பு: கோப்புப் பெயர்களுக்கிடையே காற்புள்ளி இடக் கூடாது. வெற்று இடவெளி/இடவெளிகள் மூலமாக அவற்றைப் பிரிக்க வேண்டும்.

அடுத்து `sort` என்னும் பயனுள்ள கட்டளை ஒன்று உண்டு. குறிப்பிடும் கோப்பிலுள்ள வரிகளை அகர வரிசையில் வரிசைப்படுத்தி, வரிசைப்படுத்திய விவரங்களை அடிப்படை வெளியீட்டுக்கு அனுப்பிவைக்கும். வரிசைப்படுத்திய வெளியீட்டை `more`, `cat -n`, `lpr` அல்லது இவற்றின் பொருத்தமான சேர்மானக் கட்டளைகளுக்கு அனுப்பிவைக்கமுடியும்.

எடுத்துக்காட்டுகள்:

\$ sort file3 | more

\$ sort file3 | cat -n | more

\$ sort file3 | cat -n | lpr

`tee` கட்டளைபற்றி ஏற்கெனவே அறிந்துள்ளோம். அது இரட்டைப் பணிகளை ஆற்றுகிறது.

tee கட்டளை அடிப்படை வெளியீட்டை ஒரு கோப்பில் பதிவு செய்கிறது. அடிப்படை வெளியீட்டை எழுதவேண்டிய புதிய கோப்பின் பெயரைச் செயலுருபாக (argument) ஏற்றுக்கொள்கிறது. அடிப்படை வெளியீடு கிடைக்கப்பெற்றவுடன், tee கட்டளை அதனை இரு நகல்களாகப் பிரிக்கிறது. பெரும்பாலும் அதில் ஒரு நகல், tee கட்டளைக்கு அடுத்ததாக இடம்பெறும் கோப்புக்குத் திருப்பிவிடப்படும். மற்ற நகலை திரைக்கு அனுப்பிவைக்கும். கீழே உள்ள எடுத்துக்காட்டு, file5-ன் உள்ளடக்கத்தை file6-ல் பதிவு செய்வதுடன், அதனைத் திரையிலும் காட்டுகிறது.

\$ cat file5 | tee file6

ஒரு கோப்பின் வரிசைப்படுத்தப்பட்ட உள்ளடக்கத்தை வேறொரு கோப்பில் பதிவு செய்வதுடன், திரையிலும் காட்டச்செய்ய முடியும்.

எடுத்துக்காட்டு:

\$ sort file5 | tee sfile5

sfile5-ல் file5-ன் உள்ளடக்கம் வரிசைப்படுத்தப்பட்டுப் பதிவாகியிருக்கும். வரிசையாக்கப்பட்ட உள்ளடக்கம் திரையிலும் காட்டப்படும்.

sort கட்டளையினால், file5-ன் உள்ளடக்கம் எந்த வகையிலும் பாதிக்கப்படாது. அந்த உள்ளடக்கத்தின் நகல்தான் வரிசைப்படுத்தப்படுகிறது. file5 - ல்,

Chandran

Ashok

Malar

என மூன்றுவரித் தகவல் உள்ளதாகக் கொள்வோம்.

sort, cat, tee, lpr ஆகிய கட்டளைகளைக் கீழே உள்ளவாறு இணைக்க முடியும்:

எடுத்துக்காட்டு:

\$sort file5 | tee sfile5 | lpr

இதன் வெளியீடு:

Ashok

Chandran

Malar

மேற்கண்டவாறு திரையில் காட்டப்படுவதுடன், sfile5 கோப்பிலும் பதிவு செய்யப்படும். அத்தோடு, இதே விவரத்தின் அச்சிட்ட நகலும் உங்களுக்குக் கிடைக்கும்.

\$ sort file5 | cat -n | tee sfile5 | lpr

இதன் வெளியீடு:

1. Ashok

2. Chandran

3. Malar

இந்த விவரம் திரையில் காட்டப்படுவதுடன், **sfile5** கோப்பிலும் பதிவாகும். அத்தோடு, இதன் அச்சிட்ட நகலும் கிடைக்கும்.

tee மற்றும் **cat -n** ஆகிய கட்டளைகளை இடம் மாற்றிக்கொள்ள முடியும். வெளியீடு அதற்கேற்ப மாறும்.

கோப்பு நகலாக்கம் (Copying Files)

கோப்புகளைக் கையாள்கையில், ஒரு கோப்பின் உள்ளடக்கத்தை வேறொரு கோப்பில் நகலெடுக்க வேண்டிய தேவை அடிக்கடி ஏற்படலாம். நீங்கள் இதுவரை பார்த்த நுட்பமான கட்டளைகள் மூலம், கோப்புகளை நகலெடுப்பது சாத்தியம் என்ற போதிலும், **cp (copy)** கட்டளை மூலம் நேரடி முறையில் இப்பணியைச் செய்யமுடியும்.

கட்டளை அமைப்பு

\$ cp [தேர்வுகள்] <மூலக்கோப்பு(கள்)> <இலக்குக் கோப்பகம் / கோப்பு>

சதுர அடைப்புக்குறிகளுக்குள் இருப்பவை விருப்பத் தேர்வுகள். தேவைப்பட்டால் மட்டும் சேர்த்துக் கொள்ளலாம். அல்லது விட்டு விடலாம். கோண அடைப்புக்குறிகளுக்குள் ("**< >**") இருக்கும் சொல் / சொற்கள் கட்டாயமாகத் தரப்படவேண்டும். ஆனால், உண்மையான கட்டளையைத் தரும்போது, அச்சொல்/சொற்கள் குறிக்கும் உண்மையான கோப்பு/கோப்பகப் பெயர்களைத் தரவேண்டும்.

இப்போது, நீங்கள் **file1-**ன் உள்ளடக்கத்தை **file6** -ல் நகலெடுக்க விரும்புகிறீர்கள். அதற்கான கட்டளை இவ்வாறு இருக்கும்:

\$cp file1 file6

file1 என்பது மூலக்கோப்பு (அதாவது, தரவுகள் இங்கிருந்தே பெறப்படுகின்றன). **file6** என்பது இலக்குக் கோப்பு (அதாவது, பெறப்படும் தகவல் இதில்தான் எழுதப்படுகின்றன). மேற்கண்ட கட்டளை **file1-**ன் உள்ளடக்கத்தை எவ்வகையிலும் பாதிக்காது. **file6** என்பது ஏற்கெனவே இருக்கும் கோப்பு எனில் அதிலுள்ள உள்ளடக்கம் அழிக்கப்பட்டு **file1-**ன் உள்ளடக்கம் அதில் மேலெழுதப்படும். அதாவது **cp** கட்டளை முதலில்

இலக்குக் கோப்பினை உருவாக்கிய பின்னரே, **file1** கோப்பின் உள்ளடக்கத்தை அதில் எழுதும். ஏற்கெனவே இலக்குக் கோப்பு இருப்பின் அது அழிக்கப்பட்டு அதே பெயரில் ஒரு புதிய கோப்பு உருவாக்கப்படும். எனவே கோப்புகளை நகலெடுக்கும் பணி நடைபெறும்போது, இலக்குக் கோப்பில் ஏற்கெனவே உள்ள விவரங்கள் (அப்படியொரு கோப்பு ஏற்கெனவே இருப்பின்) அனைத்தையும் இழக்க நேரிடும் என்பதைக் கவனத்தில் கொள்க. **cp** கட்டளையில் **-i** என்னும் தேர்வைச் சேர்த்து, ஏற்கெனவே இலக்குக் கோப்பு இருப்பின், அதில் மேலெழுதும் முன்பாக (அதாவது, அக்கோப்பு அழிக்கப்பட்டு அதே பெயரில் புதிய கோப்பு உருவாக்கப்படும் முன்பாக) எச்சரிக்கைச் செய்தியைத் தரும். எனவே, நீங்கள் நகலாக்கப் பணியை நிறுத்திவிடலாம்.

எடுத்துக்காட்டு:

```
$cp -i file1 file2
overwrite file2? n$
```

கோப்புகள் நடப்புக் கோப்புகத்தில் இல்லையெனில், முழுப்பா தையைக் குறிப்பிடவேண்டும். **file1**, **file2** ஆகியவற்றுக்கு இடையே தொடுப்பு (**link**) ஏற்படுத்தவேண்டும் எனில், **-i** க்குப் பதிலாக **-l** தர வேண்டும்.

cp கட்டளையில் **-r** என்னும் தேர்வு தந்து ஒரு கோப்புகத்திலுள்ள அனைத்து உள்-கோப்புகங்களையும் சேர்த்து, சுழல் செயல்பாட்டு (**recursively**) முறையில் நகலெடுக்கலாம்.

எடுத்துக்காட்டு:

```
cp -r alpha alpha1
```

இக்கட்டளை, **alpha** என்னும் கோப்புகத்திலுள்ள அனைத்துக் கோப்புகளையும் உள்-கோப்புகங்களையும் (அவற்றிலுள்ள கோப்புகளையும்) **alpha1** கோப்புகத்தில் சுழல்முறையில் நகலெடுக்கும். இந்தச் செயலாக்கத்தில் ஏற்கெனவே இருக்கும் தரவுகளை இழந்துவிடுவோம் என்கிற ஆபத்தில்லை. **alpha1** கோப்புகம் ஏற்கெனவே இருக்கும் எனில், **alpha-** வின் அனைத்து உள்ளடக்கமும் அதில் சேமிக்கப்படும். **alpha1** ஏற்கெனவே இல்லையெனில், புதிதாக உருவாக்கப்பட்டு, அனைத்து கோப்புகளும் உள்-கோப்புகங்களும் அதில் சேர்க்கப்படும். **alpha1** கோப்புகம் நடப்புக் கோப்புகத்தினுள் உருவாக்கப்படும். சுழல் செய்கை (**recursion**) என்னும் சொல் குறித்து அஞ்சத் தேவையில்லை.

கணிதத்தில் தொடர்பெருக்கலின் (**factorial**) வரையறை தெரி

யுமல்லவா?

$$0! = 1$$

$$n! = n * (n-1)!$$

$$\text{factorial}(1) = 1 * \text{factorial}(0) = 1$$

$$\text{அதேபோல } \text{factorial}(2) = 2 * \text{factorial}(1) = 2$$

$\text{factorial}(n)$ கண்டறிய வேண்டுமெனில் $\text{factorial}(n-1)$ மதிப்பை அறியவேண்டும். அதனை n -ஆல் பெருக்கவேண்டும். இது, சுழல் செய்கைக்கு ஓர் எடுத்துக்காட்டு.

`cp` கட்டளைக்கு `-S` மற்றும் `-V` என என்னும் இரண்டு தேர்வுகள் உள்ளன. `-S` தேர்வு, குறியீட்டுத் தொடுப்பினை (Symbolic Link) உருவாக்கு கிறது. `-V` தேர்வு (verbose என்பதைக்குறிக்கும்) என்னென்ன செய்யப்படு கிறது என்பதை விவரிக்கும்.

கோப்புகளை நீக்குதல்

கோப்புகளையும் கோப்பகங்களையும் அழிப்பதற்கு `rm` கட்டளை பயன்படுகிறது. இக்கட்டளை `rmdir` கட்டளையைவிடத் திறன்மிக்கது.

எடுத்துக்காட்டு:

```
$rm file1 file2
```

இக்கட்டளை, நடப்புக் கோப்பகத்திலுள்ள `file1`, `file2` ஆகிய கோப்பு களை நீக்கிவிடும். கோப்பு(கள்) நடப்புக் கோப்பகத்தில் இல்லையெனில், முழுப்பாதையின் பெயரைத் தரவேண்டும். ஏற்கெனவே, `rmdir` கட்டளை பற்றி அறிந்துள்ளீர்கள். அழிக்க வேண்டிய கோப்பகம் காலியாக இல்லையெனில் `rmdir` கட்டளையால் பயனில்லை. அதுபோன்ற நிலைமைகளில் கோப்பகத்தை அழிக்க, `rm` கட்டளையை `-r` அல்லது `-R` (recursion என்பதைக் குறிக்கும்) தேர்வுடன் சேர்த்துக் கொடுத்தால் போதும்.

எடுத்துக்காட்டு:

```
$rm -r alpha1
```

இக்கட்டளை, `alpha1` கோப்பகத்தை அதன் உள்-கோப்பகங்கள் உட்பட ஒட்டுமொத்தமாக நீக்கிவிடும். `-i` மற்றும் `-V` தேர்வுகளை அதன் இயல்பான பொருளில் பயன்படுத்தலாம். `-f` என்கிற இன்னொரு தேர்வும் உண்டு. இதனை நீங்கள் பயன்படுத்தாமல் இருப்பதே நல்லது.

பதிலீட்டுக் குறிகளும் கோப்புப் பெயர் செயலுருபுகளும்

கோப்புகளின் பெயர்களில் ஒரு பகுதிதான் நினைவில் உள்ளது எனில் முழுப் பெயரையும் அறிந்து கொள்வதற்கு, லினக்ஸ் *, ?, [] (பட்டியலில் காற்புள்ளி சேராது) ஆகிய சிறப்புக் குறியீடுகளை வழங்கியுள்ளது. `ch` எனத் தொடங்கும் கோப்புகளை அல்லது `.c` என முடியும் கோப்புகளைப் பட்டியலிட விரும்பினால் * என்னும் சிறப்புக் குறியீடு உங்களுக்கு உதவும்.

எடுத்துக்காட்டு:

```
$ ls  
main.c fact.c swap.c char1 char2.ex doc1 doc2
```

```
$ ls ch*  
char1 char2.ex  
$ ls *.c  
main.c fact.c swap.c
```

* என்னும் சிறப்புக் குறியீடு, மீதமுள்ள அனைத்து எழுத்துகளுக்கும் பதிலீடாக அமைகிறது.

```
$ rm *
```

இந்தக் கட்டளை மிகவும் ஆபத்தானது. அனைத்துக் கோப்புகளையும் துடைத்தெறிந்துவிடும். வினாக்குறி (?), கோப்புப் பெயரில் விட்டுப்போன ஒற்றை எழுத்துக்குப் பதிலீடாகப் பயன்படுத்தப்படுகிறது.

```
$ ls char?
```

```
char1
```

குறிப்பு: வினாக்குறி, எத்தனை எழுத்து என்பதை நிர்ணயித்து விடுகிறது. `char?` என்ற பெயர் கோப்பின் பெயர் 5 எழுத்து நீளமுள்ளது என்பதைக் குறிக்கிறது. முதல் நான்கு எழுத்துகள் மாறாதவை-அதாவது `char` என்பது மாறாதது. கடைசி எழுத்து எந்த எழுத்தாகவும் இருக்கலாம். எண்ணாகவும் இருக்கலாம்.

`char2.ex` என்னும் கோப்பு காட்டப்படாது. காரணம் அதன் நீளம் 5 எழுத்துக்கும் அதிகம் ஆகும்.

சதுர அடைப்புக்குறிகள் [], கோப்பின் பெயரில் இருக்கின்ற எழுத்துத் தொகுதியைக் குறிப்பிடப் பயன்படுகின்றன. `doc` எனத் தொடங்கி 1 அல்லது

2 என முடியும் கோப்பின் பெயர்களைப் பட்டியலிட விரும்புகிறீர்கள். கீழேயுள்ள கட்டளையைத் தரவேண்டும்:

\$ ls doc[12]

doc1 doc2

இங்கே தேடவேண்டிய எழுத்துகள் (எண்கள்) 1 அல்லது 2. 12 என்று எடுத்துக்கொள்ளப்பட மாட்டாது. ஒரு வீச்சின் (range) வரம்பெல்லைகளைக் குறிப்பிட முடியும்.

எடுத்துக்காட்டு:

doc[1-5] doc[a-g]

இவ்வாறு தந்தால் doc1, doc2, doc3, doc4, doc5 ஆகிய பெயர்களில் கோப்புகள் உள்ளனவா எனத் தேடும். அதேபோல, doca, docb, docd, docd, doce, docf, docg ஆகிய பெயர்களையும் தேடும்.

உரை மற்றும் கட்டளைகளைத் திருத்துதல்

ஒரு கட்டளையைத் தந்தபின் Enter விசையை அழுத்தும் முன்பாக, கட்டளை வரியில் திருத்தங்கள் செய்ய முடியும். இடது அம்புக்குறி (அல்லது Crrl+B), வலது அம்புக்குறி (அல்லது Ctrl +F) ஆகிய விசைகளைப் பயன்படுத்திக் காட்டியை நகர்த்தலாம். Back Space (அல்லது Ctrl+H) மற்றும் Delete விசைகளைப் பயன்படுத்தி வழக்கம்போல் எழுத்துகளை அழிக்கலாம். Ctrl+U வரி முழுவதையும் அழிக்கும். ஒரே வரியில் ஒன்றுக்கு மேற்பட்ட கட்டளைகளைத் தரமுடியும். ஆனால் அரைப்புள்ளி (;) மூலம் அவற்றைப் பிரிக்க வேண்டும். ஒரேயொரு கட்டளையை ஒன்றுக்கு மேற்பட்ட வரிகளில் எழுதிச் செல்ல முடியும். ஆனால் ஒவ்வொரு வரியின் இறுதியிலும் பின்சாய்வுக் கோட்டைப் (\) பயன்படுத்த வேண்டும்.

கோப்புகளை நகர்த்தலும் பெயர் மாற்றுதலும்

mv (move) என்னும் கட்டளை இரண்டு பணிகளுக்குப் பயன்படுகிறது.

1. ஒரு கோப்பு அல்லது கோப்பகத்தை ஓர் இடத்திலிருந்து இன்னொரு இடத்துக்கு நகர்த்தலாம்.
2. ஒரு கோப்பு அல்லது கோப்பகத்தின் பெயரை மாற்றலாம்.

குறிப்பு: ஒரு கோப்பினை ஓரிடத்திலிருந்து இன்னொரு இடத்துக்கு நகர்த்துவது என்பது அதனை நகலெடுப்பதிலிருந்து மாறுபட்டதாகும். நகர்த்தலின்போது புதிதாகக் கோப்பு உருவாக்கப்படுவதில்லை.

கட்டளை அமைப்பு:

```
mv [தேர்வுகள்] < மூலம்> <இலக்கு>
```

எடுத்துக்காட்டு:

```
$mv temp temporary
```

இக்கட்டளை temp என்னும் கோப்பகத்தின் பெயரை temporary என்று மாற்றி அமைக்கும். mv கட்டளை மூலம், ஒரு கோப்பினை வேறிடத்துக்கு நகர்த்தலாம்.

எடுத்துக்காட்டு:

```
$mv file1 /home/ilamathi/personal/file1
```

இந்த வகையான கட்டளைகளில், -i, -v, -f ஆகிய தேர்வுகளைப் பயன்படுத்திக்கொள்ள முடியும்.

கணிப்பொறியின் தேதி, நேரம் காணுதல்

\$ குறியை அடுத்து date என்னும் கட்டளை கொடுத்து கணிப்பொறியின் உள்ளமைந்த தேதி மற்றும் நேரம் ஆகியவற்றைப் பார்வையிடலாம்.

எடுத்துக்காட்டு:

```
$ date
```

```
Wed July 07:11:41:12 ET 2004
```

தேதி, நேரம் ஆகியவற்றைத் திரையில் வெளியிடுவதற்கு முன்பாக அவற்றை வடிவமைத்துக்கொள்ள பல்வேறு தேர்வுகள் உள்ளன. இந்தத் தேர்வுகள் இரட்டை மேற்கோள் குறிகளுக்குள் தரப்பட வேண்டும். மேற்கோள் குறிகளுக்குள் அவை + குறியுடன் தொடங்க வேண்டும்.

தேதியில் நாள், மாதம், ஆண்டையும் நேரத்தில் மணி, நிமிடம், வினாடியையும் கண்டறிய முடியும். அதற்கான தேர்வுகள் கீழே:

தேர்வு

செயல்பாடு

%d	மாதத்தில் நாள் (இலக்கத்தில்)
%m	ஆண்டில் மாதம் (இலக்கத்தில்)
%y	ஆண்டு (கடைசி இரண்டு இலக்கங்கள்)
%D	தேதி mm/dd/yy வடிவில்
%H	மணி (00 முதல் 23 வரை)

%M	நிமிடங்கள் (00 முதல் 59 வரை)
%S	வினாடிகள் (00 முதல் 59 வரை)
%T	நேரம் HH:MM:SS வடிவில்
%a	சுருக்கப்பட்ட வார நாள் பெயர்கள் (Sun முதல் Sat வரை)
%h	சுருக்கப்பட்ட மாதப் பெயர்கள் (jan முதல் dec வரை)
%r	நேரம் AM/PM குறிமானத்தில் Date கட்டளையின் தேர்வுகள்

குறிப்பு: %m, %M ஆகியவற்றில் மிகவும் கவனமாக இருக்க வேண்டும். %m, ஆண்டிலுள்ள மாதத்தை எண்ணாகக் காட்டும். %M, நிமிடங்களை எண்ணாகக் காட்டும். அதேபோல, %h மற்றும் %H பற்றியும் கவனம் தேவை. %h சுருக்கப்பட்ட மாதப் பெயரைத் தரும். %H, மணியைக் குறிக்கும். date கட்டளையில் %D பயன்படுத்தியுள்ள எடுத்துக்காட்டைப் பார்க்க இருக்கிறீர்கள்.

எடுத்துக்காட்டு:

```
$ date "+%m"
7
$ date "+%D"
07/07/04
$ date "+%T"
11:43:14
```

குறிப்பு: மேற்கண்ட தேர்வுகளின் உதவியுடன், முறைமை நிர்வாகி, date கட்டளையின் எந்தப் பகுதியையும் மாற்ற முடியும்.

3.8 கோப்பு முறைமை: mount மற்றும் umount

லினக்ஸ் முறைமையின் அனைத்துக் கோப்புகளும் ஓர் ஒட்டுமொத்தக் கோப்பக மரத்தில் இணைக்கப்பட்டுள்ளன; நிலைவட்டுகள், நெகிழ்வட்டுகள், குறுவட்டுகள் போன்ற பல்வேறு சேமிப்புச் சாதனங்களில் அக் கோப்புகள் பதிவுசெய்யப் பட்டிருக்கலாம். ஒரு குறிப்பிட்ட சேமிப்புச் சாதனத்தில் உள்ள லினக்ஸ் கோப்புகள் ஒரு கோப்பு முறைமை (File System) ஆக ஒழுங்கமைக்கப்பட்டுள்ளன. உங்கள் லினக்ஸ் கோப்பக மரம், வெவ்வேறு சேமிப்புச் சாதனங்களில் அமைந்துள்ள பல கோப்பு

முறைமைகளைக் கொண்டிருக்கலாம். கோப்புகள் தம்மளவில், வேரிலிருந்து தொடங்கும் ஒரு முழு நிறைவான கோப்பக மரமாக ஒழுங்கமைக்கப்பட்டுள்ளன. மரத்தின் வேர், நிலைவட்டின் பாகப்பிரிவில் (Partition) உள்ள கோப்பு முறைமையில் அமைந்திருந்த போதிலும், குறுவட்டு, நெகிழ்வட்டுக் கோப்பு முறைமையிலிருக்கும் கோப்புகளுக்கு ஒரு பாதைப்பெயர் இருக்கும்.

லினக்ஸ் கோப்புகளை உள்ளடக்கிய ஒரு நெகிழ்வட்டு தனக்கே உரிய கோப்பக மரத்தைக் கொண்டிருக்கும். இந்த மரம், முதன்மை மரத்திலிருந்து பிரிக்கப்பட்ட ஒரு கிளை-மரம் (sub-tree) ஆகும். உங்கள் பள்ளியை ஒரு கிளை-மரத்துக்கு ஒப்பிடலாம். உங்கள் பள்ளி சுமுகமான முறையில் செயல்பட வேண்டுமெனில், அதனைப் பள்ளிக் கல்வி இயக்குகரகத்தோடு (Directorate of School Education) இணைக்கவேண்டும். பள்ளிக் கல்வி இயக்குநரைக் கலந்து ஆலோசிக்காமலேயே உங்கள் பள்ளி சில நடவடிக்கைகளை மேற்கொள்ள முடியும் என்றாலும் இயக்குநரகத்தோடு இணைப்பு முக்கியம். அதுபோல, நெகிழ்வட்டும் குறுவட்டும் மூலக்கோப்பகத்தையே (root-directory) முழுமையாகச் சார்ந்திருக்கின்றன. அவற்றின் கோப்பு முறைமையில் உள்ள கோப்புகளை அணுக வேண்டுமெனில், மூலக் கோப்பகத்திலிருந்தே தொடங்கவேண்டும். அதற்காக, கிளை-மரத்தை முதன்மை மரத்துடன் இணைக்க வேண்டும். அவ்வாறு இணைக்கப்படாத வரையில் நெகிழ்வட்டியிலுள்ள கோப்புகளை உங்களால் அணுக இயலாது. அனைத்துச் சேமிப்பு ஊடகங்களுக்கும் இதுபொருந்தும். அவை ஏற்கெனவே இணைக்கப்பட்டிருந்தால் ஒழிய அவற்றின் உள்ளடக்கத்தை அணுக முடியாது. நிலைவட்டின் பாகப்பிரிவில் உள்ள கோப்பு முறைமையாக இருந்தாலும் அதுவும் mount கட்டளை மூலம் நிறுவப்பட வேண்டும். ஆனால் இந்த நடவடிக்கையை இயக்க முறைமை தானே பார்த்துக் கொள்கிறது.

ஒரு சேமிப்புச் சாதனத்தின் கோப்பு முறைமைக்கும் உங்கள் முதன்மைக் கோப்பக மரத்துக்கும் இடையே இணைப்பை நிறுவும் செயல்பாடு, 'சாதனம் பொருத்துதல்' (mounting the device) என்றழைக்கப்படுகிறது. mount கட்டளை மூலம் இது நிறைவேற்றப்படுகிறது. அதன்பிறகு அச்சாதனத்தின் கோப்பகத்துக்கு மாறிக் கொண்டு அதிலுள்ள கோப்புகளை அணுகலாம். இதிலுள்ள முக்கிய குறைபாடு யாதெனில், பொருத்துகைச் செயல்பாட்டை மூலப் பயனர் (root user) மட்டுமே மேற்கொள்ள முடியும். இது, ஓர் இடையூறாகத் தோன்றினாலும், கணிப்பொறி முறைமையின் ஒருங்கமைவைப் (Integrity) பாதுகாக்கிறது. மூலப் பயனராக உள்புகும்போது கட்டளைவரியின் தூண்டுகுறி \$ லிருந்து # ஆக மாறிப்

போவதைக் கவனத்திருப்பீர்கள். எனவே, **mount** கட்டளை # குறியில் தரப்பட வேண்டும் என்பதை நினைவில் கொள்க.

இயல்பாகவே, **mount** கட்டளை இரண்டு செயலுருபுகளை (**arguments**) ஏற்கும். அவற்றுள் ஒன்று, நெகிழ்வட்டு போன்ற சேமிப்பு சாதனத்தைக் குறிக்கிறது. இதன் வழியாகவே லினக்ஸ், கோப்பு முறைமையை அணுகுகிறது. இரண்டாவது செயலுருபு, அடிப்படைக் கோப்புக் கட்டமைப்பில், இந்தப் புதிய கோப்பு முறைமை இணைக்கப்படவுள்ள கோப்பகத்தைக் குறிக்கிறது. சேமிப்புச் சாதனத்திலுள்ள கோப்பு முறைமையை, முதன்மைக் கோப்பக மரத்திலுள்ள **destination** என்னும் கோப்பகத்தில் இணைக்க விரும்புகிறீர்கள். “**device**” என்கிற சிறப்பு சாதனக் கோப்பு, உங்கள் கணிப்பொறியோடு வன்பொருள் சாதனத்தை இணைக்கிறது. **mount** கட்டளையின் தொடரமைப்பு (**syntax**) வருமாறு:

mount device destination

சாதனக் கோப்புகள் **/dev** கோப்பகங்களில் வைக்கப்பட்டுள்ளன. அவை பொதுவாகச் சுருக்கப்பட்ட பெயரைக் கொண்டிருக்கும். சாதன வரிசை எண், பெயரின் இறுதியில் இடம்பெற்றிருக்கும். எடுத்துக்காட்டாக, **fd0** என்பது உங்கள் கணிப்பொறியில் பொருத்தப்பட்டுள்ள முதலாவது நெகிழ்வட்டகத்தைக் குறிக்கும். (**fd0**- வின் கடைசி எழுத்து ஓ-அல்ல, சுழியம் (**Zero**) ஆகும்). அதுபோலவே, **fd1** என்பது, உங்கள் கணிப்பொறியில் பொருத்தப்பட்டுள்ள இரண்டாவது நெகிழ்வட்டகத்தைக் (இருக்குமாயின்) குறிக்கும். பீசிக்களில் இயங்கும் லினக்ஸ் இயக்கமுறைமையில், நிலைவட்டுப் பாகப்பிரிவுகளின் (**harddisk partitions**) பெயர்கள் **hd** எனத் தொடங்கும். அடுத்து, நிலைவட்டின் வரிசையெண், ஓர் எழுத்தால் குறிக்கப்படும். இறுதியாக, அந்த நிலைவட்டின் எத்தனையாவது பாகப் பிரிவு என்பதைக் குறிக்கும் வரிசையெண் இடம்பெறும். எடுத்துக்காட்டாக, **hda2** என்பது, முதலிலை வட்டில் இரண்டாவது பாகப்பிரிவைக் குறிக்கும். **a** என்னும் எழுத்து முதல் நிலைவட்டையும், **2** என்னும் இலக்கம் இரண்டாவது பாகப்பிரிவையும் குறித்து நிற்கிறது.

ஒரு கோப்பு முறைமையை அணுக வேண்டுமெனில் அதனை முதலில் பொருத்த வேண்டும். நெகிழ்வட்டுகளும், குறுவட்டுகளும் **mount** கட்டளை மூலம் வெளிப்படையாகப் பொருத்தப்படுகின்றன. கீழேயுள்ள கட்டளை, முதல் நெகிழ்வட்டகச் சாதனத்தில் (**fd0**) உள்ள நெகிழ்வட்டை, **/destination** என்னும் கோப்பகத்தில் பொருத்துகிறது.

mount /dev/fd0 /destination

உங்கள் கணிப்பொறியின் இயக்கத்தை நிறுத்தும் முன்பாகவோ அல்லது பொருத்தப்பட்ட ஒரு கோப்புமுறைமையை நீக்கிவிட்டு வேறொன்றை நிறுவுவதற்கு முன்பாகவோ, ஏற்கெனவே பொருத்தப்பட்ட கோப்பு முறைமையை கழற்றிவிட (unmount) வேண்டும். முன்பே கூறியபடி, உங்கள் முதன்மைக் கோப்பு முறைமை தானாகவே கழற்றப்பட்டு விடும். ஒரு நெகிழ்வட்டை நெகிழ்வட்டகத்தில் செலுத்தி, அதனை முதன்மைக் கோப்பு முறைமையோடு பொருத்தியுள்ளீர்கள். இப்போது, அந்த நெகிழ்வட்டை எடுத்துவிட்டு வேறொரு நெகிழ்வட்டைச் செருக விரும்புகிறீர்கள். முதலில் நீங்கள் முந்தைய நெகிழ்வட்டை, முதன்மைக் கோப்பு முறைமையிலிருந்து கழற்றிவிட (unmount) வேண்டும். அதன்பிறகு நெகிழ்வட்டகத்திலுள்ள நெகிழ்வட்டை நீக்கிவிட்டுப் புதிய நெகிழ்வட்டைச் செருக வேண்டும். **umount** கட்டளைமூலம் ஒரு கோப்பு முறைமையைக் கழற்றிவிடலாம். (கட்டளைச் சொல்லை எழுத்துக் கூட்டிப்பாருங்கள். **u**-க்கும் **m**-க்கும் இடையில் **n** இல்லை) **umount** கட்டளை, ஒரு சாதனத்தின் பெயர் அல்லது அச்சாதனம் பொருத்தப்பட்ட கோப்பகத்தை செயலுருபாக ஏற்கும். அதன் கட்டளை அமைப்பு:

umount device (அல்லது **destination**)

கீழேயுள்ள கட்டளைகள், **/destination** கோப்பகத்தில் பொருத்தப்பட்ட நெகிழ்வட்டைக் கழற்றிவிடும்.

umount /dev/fd0

umount /destination

umount கட்டளையில் ஒரு முக்கியமான நிபந்தனை விதி உள்ளது. நீங்கள் தற்போது பணியாற்றிக் கொண்டிருக்கும் கோப்பு முறைமையைக் கழற்றிவிட முடியாது. ஒரு மரக்கிளையின் நுனியில் அமர்ந்துகொண்டு அதன் அடிப்பகுதியை வெட்டக் கூடாது அல்லவா?

நெகிழ்வட்டினை பொருத்துதலும், வடிவமைத்தலும்

ஒரு புத்தகத்தை நீங்கள் படிக்க விரும்பினால் முதலில் அப்புத்தகம் உங்களிடம் இருக்க வேண்டும். அதுபோல, ஒரு நெகிழ்வட்டின் உள்ளடக்கத்தை அணுக விரும்பினால், முதலில் அதனைப் பொருத்தியிருக்க (**mount**) வேண்டும். ஏற்கெனவே சொன்னபடி, **/dev/fd0** என்பது நெகிழ்வட்டகத்தைக் குறிக்கும். நீங்கள் விரும்பியப்படி ஏதேனும் ஒரு கோப்பகத்தில் இதனைப் பொருத்திக் கொள்ளலாம். என்றாலும் லினக்ஸ் முறைமை, நெகிழ்வட்டுகளை பயன்படுத்துவதற்கென **/mnt/floppy** என்னும் வசதியான கோப்பகத்தை ஏற்கெனவே உருவாக்கி வைத்துள்ளது.

(mnt என்ற சொல் mount என்பதைக் குறிக்கிறது). கீழேயுள்ள கட்டளை நெகிழ்வட்டினை முறைமையில் பொருத்திவிடும்.

```
#mount /dev/fd0 /mnt/floppy
```

நெகிழ்வட்டிலுள்ள கோப்புகளைப் படிக்கக் கணிப்பொறி முயலும். முதல் நெகிழ்வட்டை முறைப்படி கழற்றாமல் (unmount), நெகிழ்வட்ட கத்திலுள்ள வட்டினை மாற்றினால், உங்களுக்கு ஒரு பிழைசுட்டும் செய்தி கிடைக்கும். நெகிழ்வட்டை மாற்ற விரும்பினால், dev/fd0-ல் முதல் வட்டினைக் கழற்றியபின், புதிய நெகிழ்வட்டை கீழேயுள்ளவாறு வெளிப்படையாக நிறுவ வேண்டும்.

```
#umount /dev/fd0
```

அல்லது

```
# umount /mnt/floppy
```

பிறகு

```
# mount /mnt/floppy
```

குறிப்பு: mkfs (make formattings) கட்டளை நெகிழ்வட்டினை வடிவமைக்கும்.

சிடி-ரோம்களைப் பொருத்துதல் (Mounting CD-Roms)

சிடி-ரோம் வட்டினைப் பொருத்த விரும்பினால், சாதனப் பெயரைக் குறிப்பிட வேண்டியதில்லை. லினக்ஸ் முறைமையில், சிடி-ரோம் கோப்பு முறைமைகளைக் கையாள்வதற்கென /mnt/cdrom என்னும் கோப்பகம் உள்ளது. கீழே உள்ள கட்டளை சிடி-ரோம் வட்டினைப் பொருத்தும்:

```
# mount /mnt/cdrom
```

சிடி-ரோம் வட்டினை மாற்ற விரும்பினால், இப்போது இருக்கும் சிடி-ரோம் வட்டினைக் கழற்றிவிட்டு, புதிய சிடி-ரோம் வட்டினைப் பொருத்தவேண்டும். அதற்கான கட்டளைகள்:

```
# umount /mnt/cdrom
```

(இப்போது சிடி-ரோம் வட்டினை மாற்றிக் கொள்ளவும்)

```
# mount /mnt/cdrom
```

வேறொரு கோப்பகத்தில் சிடி-ரோம் வட்டினைப் பொருத்த விரும்பினால், mount கட்டளையில் சாதனப் பெயரைச் சேர்த்துக் கொள்ள வேண்டும். கீழேயுள்ள கட்டளை சிடி-ரோம் வட்டகத்தில் குறுவட்டினை

/destination என்னும் கோப்பகமாகப் பொருத்தும். இந்த எடுத்துக்காட்டில், சிடி-ரோமுக்குரிய சாதனப் பெயர் /dev/hdc ஆகும்.

mount /dev/hdc /destination

தொகுப்புரை:

கோப்புகளைக் கையாள்வதற்கான சில பொதுவான கட்டளைகள்:

- cat** - கோப்புகளின் உள்ளடக்கத்தைக் காட்டும்.
- more** - குறிப்பிடும் கோப்பின் உள்ளடக்கத்தைப் பக்கம் பக்கமாக நிறுத்திக் காண்பிக்கும். f மற்றும் b விசைகளை அழுத்தி, முறையே முன்னோக்கி, பின்னோக்கி நகரலாம்.
- tee** - கோப்பின் உள்ளடக்கத்தை திரையில் காட்டும். அதேவேளையில் அதனை இன்னொரு கோப்பில் நகலெடுக்கும்.
- | (pipe)** - ஒரு கட்டளையின் வெளியீட்டை இன்னொரு கட்டளைக்கு உள்ளீடாக அனுப்பிவைக்கும்.
- > (output redirection)** - தரவுகளைக் கோப்புக்குத் திசைதிருப்பும்.
- < (input redirection)** - கோப்பிலுள்ள தரவுகளைக் கட்டளைக்குத் திசைதிருப்பும்.
- cp** - கோப்புகளை நகலெடுக்கும்.
- rm** - ஒரு கோப்பு அல்லது கோப்பகத்தை நீக்கும்.
- mv** - கோப்புகள் அல்லது கோப்பகங்களை நகர்த்தும் அல்லது பெயர்மாற்றும்.
- mount** - ஒரு சேமிப்புச் சாதனத்திலுள்ள கோப்பு முறைமையை முதன்மைக் கோப்பக மரத்துடன் பொருத்தும். இக்கட்டளை இரு செயலுருபுகளை ஏற்கும்.
- umount** - பொருத்தப்பட்ட ஒரு கோப்பு முறைமையைக் கழற்றும்.
- fd0** - உங்கள் கணினிப்பொறியில் பொருத்தப்பட்டுள்ள முதல் நெகிழ்வட்டகத்தைக் குறிக்கும்.

மேலும் சில கட்டளைகள்:

- date** - தற்போதைய கணிப்பொறித் தேதியையும், நேரத்தையும் பார்க்கவும், மாற்றவும் பயன்படும்.
- tput clear** - திரையிலுள்ள விவரங்களைத் துடைக்கும்.
- tput cup** - குறிப்பிட்ட கிடக்கை, நெடுக்கையில் காட்டியை நிறுத்தும்.
- man** - எந்தவொரு லினக்ஸ் கட்டளைபற்றியும் உதவிச் செய்திகளை வழங்கும்.

பயிற்சி வினாக்கள்:

I. புள்ளிட்ட இடங்களை நிரப்புக

1. _____ கட்டளை மூலம் கோப்புகளை நேரடியாக நகலெடுக்கலாம்.
2. ஒரு குறிப்பிட்ட கட்டளை பற்றிய உதவிச் செய்திகளை அறிய விரும்பினால், _____ கட்டளை அத்தகைய உதவியைத் தரும்.
3. ஒரு பெரிய கோப்பின் உள்ளடக்கத்தை _____ கட்டளை, பக்கம் பக்கமாக நிறுத்திக் காண்பிக்கும்.
4. cp கட்டளையில் _____ தேர்வு, இலக்குக் கோப்பில் மேலெழுதப்படுவது பற்றி எச்சரிக்கை செய்யும்.
5. \$cat file1 _____ file2 என்னும் கட்டளை file1- ன் உள்ளடக்கத்தை file2- யின் பின்சேர்க்கும்.
6. திசைதிருப்பல் செயல்பாடு மூலமாக ஏற்கெனவே இருக்கும் கோப்பில் மேலெழுதப்படுவதை தவிர்க்க _____ என்னும் வசதியைப் பயன்படுத்தலாம்.
7. கோப்பு இறுதிக் குறியீடு _____ ஆகும்.
8. ஒரு சாதனத்தைப் பொருத்துதல் (mounting) என்பது ஒரு சேமிப்பு சாதனத்திலுள்ள கோப்பு முறைமை, முதன்மைக் கோப்பக மரத்துக்கும் இடையே இணைப்பை _____ ஆகும்.
9. நிலைவட்டின் பாகப்பிரிவில் உள்ள கோப்பு முறைமையில் மரத்தின் வேர் இருந்தாலும், சிடி-ரோம் கோப்பு முறைமையில் இருக்கும் கோப்புகளுக்காக ஒரு _____ இருக்கும்.

10. _____ பொருத்தும் பணியை மட்டுமே செய்யும்.
11. லினக்ஸ் முறைமையில் உள்ள அனைத்துக் கோப்புகளும் ஓர் ஒட்டுமொத்த _____ மரத்தில் இணைக்கப்பட்டுள்ளன.
12. லினக்ஸ் கோப்புகளைக் கொண்ட ஒரு நெகிழ்வட்டு தனக்கே உரிய கோப்பக _____ தைக் கொண்டிருக்கும்.

II. சரி அல்லது தவறு எனக் குறிப்பிடுக

1. ls கட்டளை, கோப்புகளின் உள்ளடக்கத்தைக் காட்டும்.
2. mv கட்டளை, கோப்பினை ஓரிடத்திலிருந்து வேறிடத்துக்கு நகர்த்தும்.
3. tput cup 20 20 என்னும் கட்டளை, சி-மொழி நிரலாக்கத்தில் gotoxy(20, 20) என்னும் கட்டளைக்கு இணையானது.
4. echo -n "This is nice" என்னும் கட்டளை, This is nice எனத் திரையில் காட்டி, காட்டியை (cursor) அடுத்த வரிக்கு எடுத்துச் செல்லும்.
5. read myname என்னும் உரைநிரல் கட்டளை சி-மொழி நிரலாக்கத் திலுள்ள gets(myname) கட்டளைக்கு இணையானது.
6. cat கட்டளை ஒரு கோப்பின் உள்ளடக்கத்தைப் பக்கம் பக்கமாக நிறுத்திக் காண்பிக்கும்.
7. \$ cat file1 > file1 என்னும் கட்டளை file1- ன் உள்ளடக்கத்தை அதன் மீதே எழுதிக்கொள்ளும்.
8. \$ cat file1 > file2 என்னும் கட்டளை, file1, file2 ஆகிய இரண்டு கோப்புகளும் இருந்தால் மட்டுமே செயல்படும்.
9. lpr கட்டளை அடிப்படை வெளியீட்டை உள்ளீடாக எடுத்துக் கொண்டு, அதனைத் திரைக்கு அனுப்பிவைக்கும்.
10. mount கட்டளை இரண்டு செயலுருபுகளை ஏற்கும்.
11. umount கட்டளை, ஒரு சாதனத்தின் பெயரையோ, அது பொருத் தப்பட்ட கோப்பகத்தின் பெயரையோ செயலுருபாக ஏற்கும்.

III. விடை தருக

1. rm -r, rmdir ஆகிய இரு கட்டளைகளுக்கும் உள்ள வேறுபாடு யாது?
2. My name is <yourname> என்பதுபோன்ற செய்தியைத் திரையில் காட்டச் செய்வது எப்படி?

3. ஒரு கோப்பகத்தை அதிலுள்ள உள்-கோப்பகங்களோடு ஒட்டு மொத்தமாக அழிப்பது எவ்வாறு?
4. cat கட்டளை என்ன செய்கிறது? cat கட்டளையின் அனைத்து வடிவங்களையும் எழுதி விவாதிக்க.
5. குழாய்செலுத்தல், திசைதிருப்பல் - வேறுபடுத்திக் காட்டுக.
6. mv, cp கட்டளைகளை வேறுபடுத்திக் காட்டுக.
7. file1 -ன் உள்ளடக்கத்தை file2- ல் நகலெடுப்பதற்குரிய வெவ் வேறான வழிமுறைகளை விளக்குக.
8. ஒரு கோப்பகத்தை அதிலுள்ள அனைத்துக் கோப்புளுடனும் நகலெடுப்பது எவ்வாறு?

3.9 விஜ தொகுப்பி (vi Editor)

பெரும்பாலும் கோப்புகளை உருவாக்கவும், அழிக்கவும், திருத்தித் தொகுக்கவும் தொகுப்பிகள் (Editors) பயன்படுகின்றன. லினக்ஸில் பல தொகுப்பிகள் உள்ளன. அவற்றுள் சில அதிநுட்பம் வாய்ந்தவை. அனைத்து லினக்ஸ் முறைமைகளிலும் இரண்டு அடிப்படையான தொகுப்பிகள் உள்ளன. அவை எட் (ed), விஜ (vi) தொகுப்பிகள் ஆகும். எட் ஒரு நேரத்தில் ஒரு வரியைத் திருத்த மட்டுமே பயனரை அனுமதிக்கும். எனவே அவ்வளவாகப் பயன்படுத்தப்படுவதில்லை. விஜ தொகுப்பியில் ஒரு நேரத்தில் திரையில் ஒரு பக்கம் வரையிலான உரையைத் திருத்தியமைக்க முடியும். எனவே, விஜ தொகுப்பியே இப்போதும் பரவலாகப் பயன்படுத்தப்படுகிறது. யூனிக்ஸில் விஜ தொகுப்பி அறிமுகப்படுத்தப்பட்ட காலகட்டத்தில், பிறதொகுப்பிகள் எதிலும் இல்லாத ஏராளமான வசதிகளை, விஜ தொகுப்பி வழங்கியது.

தொகுப்பிகள் விசைப்பலகையை முற்றிலும் வேறான இரு பணிகளுக்குப் பயன்படுத்திக் கொள்கின்றன. அவை:

- 1) தொகுப்புக் கட்டளைகளைக் குறிப்பிட.
- 2) எழுத்துகளின் உள்ளீட்டை ஏற்றுக் கொள்ள.

பொதுவான பீசி தொகுப்பிகள், மேற்கண்ட இருபணிகளையும் விசைப்பலகையின் விசைகளுக்கிடையே பிரித்துக் கொள்கின்றன. உரையை உள்ளிட எழுத்து விசைகளையும், தொகுப்புக் கட்டளைகளுக்குப் பணி விசைகளையும் (Functional Keys), கட்டுப்பாட்டு விசைகளையும் (Control Keys) பயன்படுத்திக் கொள்கின்றன. அத்தகைய பீசி தொகுப்பிகள் விரிவாக்கப்

பட்ட விசைப்பலகைகளைப் பெரிதும் சார்ந்திருக்கின்றன. விசைகளின் எண்ணிக்கை நாளுக்கு நாள் அதிகரித்துக் கொண்டே செல்கிறது.

ஆனால், யூனிக்ஸ், அதைத் தொடர்ந்து லினக்ஸ், இரண்டுமே பல வகையிலும் சிக்கனமானவை. லினக்ஸ் முறைமையில் எந்தவகை விசைப்பலகையையும் பயன்படுத்தலாம். லினக்ஸிலுள்ள தொகுப்பிகள், அகர்-வரிசை எழுத்து விசைகளுடன் மிகக் குறைந்த எண்ணிக்கையிலான பிற விசைகள் உள்ள விசைப்பலகைகளுக்கென வடிவமைக்கப்பட்டவை. சில கட்டுப்பாட்டுக் குறியீடுகள், **Esc** விசை மற்றும் **Enter** விசை - அவ்வளவுதான். மிகக்குறைந்த விசைகளை வைத்துக் கொண்டு, விஐ தொகுப்பி, நிலைமையை எவ்வாறு சமாளிக்கிறது எனப் பார்ப்போம்.

நீங்கள், உங்கள் பள்ளியில் மாணவராக நடந்து கொள்கிறீர்கள். அதே வேளையில் உங்கள் வீட்டில் மகனாகவும்/மகளாகவும் நடந்து கொள்கிறீர்கள். பள்ளியில் இருக்கும்போது சில வரைமுறைகளுக்கு உட்பட்டுச் செயல்படுகிறீர்கள். ஆனால் வீட்டில் இருக்கும்போதோ சுதந்திரமாகச் செயல்படுகிறீர்கள். ஒரே நபர் இரட்டை வேடம் (சொல்லப் போனால் பல வேடங்கள்) பூணுகிறார். இதேபோலத்தான், விஐ தொகுப்பி, விசைப்பலகைகளை இரட்டைவேடம் பூணச் செய்கிறது. விஐ தொகுப்பியில் இருவகைப் பாங்குகள் உள்ளன. அவை:

1) கட்டளைப் பாங்கு (Command mode)

2) உள்ளீட்டுப் பாங்கு (Input mode)

கட்டளைப் பாங்கினில், விசைப்பலகையிலுள்ள அனைத்து விசைகளும் தொகுப்புக் கட்டளை (Edit Command) விசைகளாக மாறிவிடுகின்றன. உள்ளீட்டுப் பாங்கினில், விசைப்பலகை ஒரு சாதாரண தட்டச்சுப் பொறிபோலச் செயல்படுகிறது.

விசைப்பலகையின் பாங்கினை மாற்றும்போது, அதன் செயல்தன்மையும் மாறிவிடுகிறது. எடுத்துக்காட்டாக, கட்டளைப் பாங்கினில், x என்னும் விசை, சாதாரண விசைப்பலகையில், delete விசை செயல்படுவது போல, காட்டி (cursor) இருக்கும் இடத்திலுள்ள எழுத்தை அழித்துவிடும். ஆனால், உள்ளீட்டுப் பாங்கினில், அதே x விசை காட்டி இருக்கும் இடத்தில் x என்னும் எழுத்தைச் சேர்க்கும். Esc விசை மட்டும் விதிவிலக்கானது. Esc விசையை அழுத்தினால், கட்டளைப் பாங்குக்கு மாறிவிடும். ஆனால், ஏற்கெனவே கட்டளைப் பாங்கினில் இருப்பின், Esc விசை 'பீப்' ஒலி எழுப்பும். விஐ தொகுப்பியைப் புதிதாகப் பயன்படுத்துபவர்களுக்கு இந்தச் செயல்பாடு மிகவும் உதவியாக அமைகிறது. எந்தப் பாங்கினில்

பணியாற்றுகிறீர்கள் என்கிற ஐயம் ஏற்பட்டால், விடுபடு விசையை அழுத்தவும். 'பீப்' ஒலி கேட்டால், கட்டளைப் பாங்கினில் இருப்பதை அறிந்து கொள்ளலாம். 'பீப்' ஒலி கேட்காவிடில், ஏற்கெனவே (விடுபடு விசையை அழுத்தும் முன்பாக) உள்ளீட்டுப் பாங்கினில் இருந்துவிட்டு இப்போது கட்டளைப் பாங்குக்கு வந்துவிட்டீர்கள் என்று பொருள். எனவே, Esc விசை பற்பல குழப்பங்களைத் தவிர்க்கிறது.

விஜ தொகுப்பியின் கட்டளைப் பாங்கு, பல தொகுப்புச் செயல்பாடுகளைச் செயல்திறனுடன் கையாள்கிற போதிலும், கோப்பினைச் சேமிப்பது போன்ற செயல்களை நிறைவேற்ற முடியாது. இப்பணிகளை, வரித் தொகுப்புக் (line-editing) கட்டளைகள் செய்து முடிகின்றன. கட்டளைப் பாங்கினில் இருக்கும்போது முக்காற்புள்ளி (:colon) வரித்தொகுப்புப் பாங்குக்கு இட்டுச் செல்லும். தேவையான செயல்பாடுகளை நிறைவேற்றிய பிறகு, Enter விசையை அழுத்தியவுடன், திரும்பவும் கட்டளைப் பாங்குக்கே வந்துவிடுவீர்கள். ஏற்கெனவே நீங்கள் அறிந்துள்ள பாங்குகளோடு, வரித் தொகுப்புப் பாங்கினையும் இப்போது அறிந்து கொண்டீர்கள்.

விஜ தொகுப்பியில் ஒருகோப்பினை உருவாக்குதல், சேமித்தல், தொகுத்தல், வெளியேறுதல்:

student என்ற பெயரில் ஏற்கெனவே உள்ள ஒரு கோப்பினைத் திருத்தவோ, அல்லது அப்பெயரில் ஒரு புதிய கோப்பினை உருவாக்கவோ விரும்பினால்,

\$ vi student

என்று கட்டளை தரவும்.

அக்கோப்பு ஏற்கெனவே இருப்பின் அதன் உள்ளடக்கம் பக்கம் பக்கமாகத் திரையில் காட்டப்படும். அப்படியொரு கோப்பு இல்லையெனில், ஒரு வெற்றுத் திரையையும், இடப்பக்கத்தில் தொடர்ச்சியான நெளிவுக் (~) குறிகளையும் காணலாம். கோப்பு இருக்கிறதோ இல்லையோ, நீங்கள் கட்டளைப் பாங்கினில் இருப்பீர்கள். திரையில் நெளிவுக் குறிகள் உள்ள பகுதி, கோப்பின் அங்கமாக இல்லை என்பதை அக்குறிகள் உணர்த்துகின்றன.

கோப்பில் விவரங்களை உள்ளிட விரும்பினால், உள்ளீட்டுப் பாங்குக்கு மாறிக் கொள்ளவேண்டும். a, i, o ஆகிய விசைகளுள் ஒன்றை அழுத்தி, கட்டளைப் பாங்குக்கு மாறிக் கொள்ளலாம். பிறகு, பெரிய எழுத்தில் இடவெளியின்றி ZZ எனப் பதியவும். CapsLock Key செயல்பாட்டில் இல்லாதபோது, Shift விசையை அழுத்திக்கொண்டு இருமுறை Z

விசையை அழுத்தவேண்டும்). இச்செயல், கோப்பினைச் சேமிக்கும்; விஐ தொகுப்பியை விட்டு வெளியேறி லினக்ஸ் செயல்தளத்துக்குத் திரும்புவீர்கள்.

கட்டளைப் பாங்கினில் இருக்கும்போது, கேப்பினைச் சேமித்து விட்டு வெளியேற வேண்டும். ஆனால், கோப்பினில் விவரங்களை உள்ளிடும் போது, அதனை அடிக்கடி சேமிக்க வேண்டும். Esc விசையை அழுத்திக் கட்டளைப் பாங்குக்கு மாறவேண்டும். அதன்பிறகு முக்காற்புள்ளி (:) அழுத்தி, வரித் தொகுப்புப் பாங்குக்குச் செல்லவேண்டும். அடுத்து, w விசையை அழுத்தவேண்டும். இந்த வரிசையிலான செயல்கள், கோப்பினைச் சேமிக்கும்; நீங்கள் கட்டளைப்பாங்குக்குத் திரும்புவீர்கள். :w கட்டளையுடன் சேர்த்து ஒரு கோப்புப் பெயரைத் தந்தால் அப்பெயரில் கோப்பு சேமிக்கப்படும். பிற சொல்செயலிகளில் (Word Processors) Save As கட்டளை செயல்படுவதுபோல இது செயல்படுகிறது.

பெயர் கொடுக்காமலே ஒரு கோப்பினை உருவாக்க முடியும். அதற்கான கட்டளை:

\$vi

இவ்வாறு பெயரின்றி ஒரு கோப்பினை உருவாக்க முயலும்போது, உண்மையில் ஒரு கோப்பு உருவாக்கப்படுவதில்லை. நீங்கள் பதிவு செய்யும் விவரங்கள் இடையகத்தில் (Buffers) சேமிக்கப்படும். தொடர்ந்து விவரங்களை உள்ளிட முடியும். இடையகத்தில் உள்ள விவரங்களைத் திருத்தியமைக்க முடியும். ஆனால் ZZ கட்டளை மூலம் கோப்பினைச் சேமிக்க முடியாது. இருக்கின்ற கோப்பினையே ZZ கட்டளை சேமிக்கும். ZZ பயன் தராது என்றபோதிலும், :w <கோப்புப் பெயர் > கட்டளை மூலம் விவரங்களைச் சேமிக்கமுடியும். இக்கட்டளை, இடையகத்தில் உள்ள விவரங்களை, நீங்கள் தரும் பெயரில் சேமிக்கும். கோப்புப் பெயர்களுக்கிடையே மோதல் (ஒரே பெயரில் இரு கோப்புகள்) ஏற்றுக் கொள்ளப்படமாட்டாது. ஏற்கெனவே இருக்கும் ஒரு கோப்பின் பெயரைத் தந்தால், அது புறக்கணிக்கப்படும். நீங்கள் வேறொரு பெயரைத் தரவேண்டும். விஐ தொகுப்பியைவிட்டு வெளியேற விரும்புகிறீர்கள். :f எனக்கட்டளை தந்தால், செயல்தளத்துக்குத் (shell) திரும்புவீர்கள். ஆனால், கோப்பினில் சில திருத்தங்கள் செய்திருப்பின் இக்கட்டளை செயல்படாது. அப்போது :q! கட்டளை கைகொடுக்கும். கோப்பினில் செய்யப்பட்ட மாற்றங்களைச் சேமிக்காமலே விஐ தொகுப்பியைவிட்டு வெளியேற உதவும்.

உரைகளுக்கு நடுவே நகர்ந்துசெல்ல அம்புக்குறி விசைகளைப் பயன்படுத்தலாம். ஆனால் அந்தப் பணிக்கும் அகரவரிசை எழுத்து விசை

களையே பயன்படுத்த விரும்பினால் அதுவும் சாத்தியமே. h,j,k,l விசைகள் உங்கள் விருப்பத்தை நிறைவேற்றும்.

h = ← (இடது அம்புக்குறி)

l = → (வலது அம்புக்குறி)

j = ↑ (கீழ் அம்புக்குறி)

k = ↑ (மேல் அம்புக்குறி)

h, j, k, l ஆகிய விசைகள் விசைப்பலகையின் நடுக் கிடக்கையில், உங்களின் வலக்கைப்புறம் உள்ளன. நான்கு விசைகளுள் h, இடது ஓரத்திலும், l வலது ஓரத்திலும் உள்ளன. எனவேதான், h விசை இடது அம்புக்குறிக்கு மாற்றாகவும், l விசை வலது அம்புக்குறிக்கு மாற்றாகவும் பயன்படுத்தப்படுகின்றன. Enter விசையை h-க்குப் பதிலாகவும், Spacebar விசையை l-க்குப் பதிலாகவும் பயன்படுத்த முடியும்.

உரையின் ஒவ்வொரு வரியும் திரையின் இடது ஓரத்தில் தொடங்கும். Enter விசையை அழுத்தும்போது வரி முடிவுபெறுகிறது. வரியின் முடிவுக்கும் திரையின் முடிவுக்கும் இடைப்பட்ட இடவெளி (Space) 'வெற்று இடவெளி' (dead space) எனப்படும். கோப்பினில் இது சேமிக்கப்படாது. h மற்றும் l விசைகளின் உதவியுடன் ஒரு வரிக்குள் அங்குமிங்கும் நகரலாம். காட்டி (cursor) ஒரு வரியின் இறுதியில் இருக்கும்போது, அடுத்த வரியின் இறுதிக்கு நகர விரும்பினால், j விசையை அழுத்தவும். முந்தைய வரியின் இறுதிக்குச் செல்ல விரும்பினால் k விசையை அழுத்தவும். உரையினூடே திரை முழுவதையும் தாண்டிச் செல்லவும் முடியும். Ctrl+F விசைகளை ஒருசேர அழுத்தினால் ஒரு திரைப்பக்கம் தாண்டி முன்னால் (Forward) செல்லலாம். Ctrl+B விசைகளை ஒருசேர அழுத்த, ஒரு பக்கம் பின்னோக்கி (Backward) நகரலாம்.

வரி எண் G

விஜ தொகுப்பி உரையின் ஒவ்வொரு வரிக்கும் வரிசையெண் இட்டு வைக்கும். வரியெண் அதைத் தொடர்ந்து G (பெரிய எழுத்து) அழுத்திக் குறிப்பிட்ட வரிக்குச் செல்லமுடியும். கோப்பின் இறுதிக்குச் செல்ல விரும்பினால், வரியெண் இடாமல் G மட்டும் அழுத்தினால் போதும்.

சொல் மடிப்பு ஓரத்தை நிர்ணயிக்க விரும்பினால், :set wm = < நெடுக்கை > என்ற கட்டளை தரவேண்டும். < நெடுக்கை > என்னுமிடத்தில் பொருத்தமான எண் இடப்பட வேண்டும்.

a விசையை அழுத்தினால், விஜ தொகுப்பி, உள்ளீட்டுப் பாங்குக்கு

மாறும். உள்ளிடும் எழுத்துகள், காட்டி இருக்குமிடத்துக்கு அடுத்த எழுத்து களாக இடம்பெறும். i விசையை அழுத்தி உள்ளீட்டுப் பங்குக்கு மாறினால், காட்டி இருக்குமிடத்துக்கு முந்தைய எழுத்துகளாகப் பதிவாகும். சிறிய எழுத்து o (ஓ) விசையை அழுத்தினால், காட்டி இருக்குமிடத்துக்குக் கீழே ஒரு வெற்று வரியை உருவாக்கும். புதிய வரியின் தொடக்கத்தில் உள்ளீடு பதிவாகுமாறு, உள்ளீட்டுப் பாங்கு அமையும்.

அழித்தல் (Deletion)

கட்டளைப் பாங்கினில், x விசை, ஒற்றை எழுத்தை அழிக்கும். ஏற் கெனவே கூறியபடி, x விசை சாதாரண விசைப்பலகையின் Delete விசை யைப்போலச் செயல்படும். x விசையை திரும்பத் திரும்ப பயன்படுத்தி, எத்தனை எழுத்துகளை வேண்டுமானாலும் அழிக்கலாம்.

dd கட்டளை, தற்போது காட்டி இருக்கும் வரியை முழுவதுமாக அழித்துவிடும். x கட்டளை அல்லது dd கட்டளைக்கு முன்னால் ஓர் எண் தந்தால், அந்த எண்ணிக்கையில் எழுத்துகளையோ வரிகளையோ அழித்து விடும்.

காட்டி இருக்கும் இடத்திலிருந்து 5 எழுத்துகளை அழிக்க விரும்பு கிறீர்கள். கட்டளைப் பாங்கினில் இருந்துகொண்டு, 5x என்று கட்டளை தரவேண்டும். காட்டி இருக்கும் வரியை அழிக்க விரும்பினால், கட்டளைப் பாங்கினில் இருந்துகொண்டு dd எனக் கட்டளை தரவேண்டும். அந்த வரி அழிக்கப்பட்டு விடும். 5dd எனக் கட்டளை தந்தால் அந்தவரி தொடங்கி 5 வரிகள் அழிக்கப்பட்டுவிடும்.

செய்தது தவிர் (Undo)

U என்னும் கட்டளை, கடைசியாகச் செய்த திருத்தத்தைத் தவிர்த்து விடும்.

வரியை முறித்தல் (Break a Line)

ஒரு வரியை முறிக்க வேண்டுமெனில், உள்ளீட்டுப் பாங்கினில் நுழைந்து, Enter விசையை அழுத்த வேண்டும். இரண்டு வரிகளை ஒட்ட வைக்க J (பெரிய எழுத்து) விசையை அழுத்தவும்.

நகர்த்துதல் (Moving)

உரையின் ஒரு பகுதியை (சில வரிகளை) நகர்த்த விரும்புகிறீர்கள். ndd (நகர்த்த விரும்பும் வரிகளின் எண்ணிக்கையை n குறிக்கிறது) கட் டளை மூலம் அந்த வரிகளை அழிக்க வேண்டும். இக்கட்டளை, விண் டோஸ் எக்ஸ்பீ புரொஃபஷனில் உள்ள வெட்டும் (cut) செயல்பாட்டை

ஓத்தது. அழிக்கப்பட்ட வரிகள் இடையகத்தில் இருத்தப்படும். இங்கே இடையகம் என்பது விண்டோஸ் எக்ஸ்பீ புரொஃபஷனில் இருக்கும் பிடிபல கையை ஓத்தது. அடுத்து, வெட்டிய வரிகளை நகர்த்த வேண்டிய இடத்தில், காட்டியை நகர்த்திக் கொள்ளுங்கள். p என்னும் விசையை அழுத்துங்கள். காட்டி இருக்கும் வரிக்கு அடுத்த வரியில், வெட்டிய வரிகள் ஓட்டவைக்கப்படும். இக்கட்டளை, விண்டோஸ் எக்ஸ்பீ புரொஃபஷனில் உள்ள ஓட்டும் (Paste) செயல்பாட்டுக்கு இணையானது.

நகலெடுத்தல் (Copying)

ஒரு வரியை yy கட்டளை மூலம் நகலெடுக்க முடியும். காட்டி இருக்கும் வரி தொடங்கி n வரிகளை நகலெடுக்க விரும்பினால், nyy எனக் கட்டளை அமைக்கவேண்டும். இக்கட்டளை அந்த வரிகளை நகலெடுத்து இடையகத்தில் இருத்தும். இடையகத்தில் இருத்தப்பட்ட வரிகளை விரும்பும் இடத்தில் ஓட்டவைக்கலாம். அந்த வரிகளை ஓட்ட வைக்க விரும்பும் இடத்துக்குக் காட்டியை நகர்த்தியபின், p விசையை அழுத்தினால் போதும். உங்கள் விருப்பம் நிறைவேறும். நகர்த்தலுக்கும், நகலெடுத்தலுக்கும் இடையே உள்ள வேறுபாட்டை விண்டோஸ் எக்ஸ்பீ-யிலேயே அறிந்துள்ளோம். nyy கட்டளையில் நகலெடுக்கும் வரிகள் அழிக்கப்படுவதில்லை.

தேடுதல் (Searching)

உரைப்பகுதிக்குள் சொல் அல்லது தொடர்களைத் தேடிக் கண்டறிய முடியும். ஒரு குறிப்பிட்ட சொல் அல்லது தொடர், உரைப்பகுதிக்குள் எங்கெல்லாம் இடம் பெற்றுள்ளது என்பதைக் கண்டறிய விரும்பினால் / அல்லது ? குறியைப் பயன்படுத்த வேண்டும். முன்சாய்வுக்கோடு (/), காட்டிக்கு அடுத்துவரும் உரைப்பகுதியில் குறிப்பிடும் தோரணியைத் (pattern) தேடிக் கண்டறிய உதவுகிறது. வினாக்குறி (?)யைப் பயன்படுத்தி பின்னோக்கித் தேடலாம். / விசையை அழுத்தியதும், திரையின் அடிப் பகுதியில் ஒருவரி தோன்றும். வரியின் தொடக்கத்தில் / குறி இருக்கும். காட்டி அடுத்துக் காத்திருக்கும். சொல் அல்லது தோரணியை உள்ளிட்டு Enter விசையை அழுத்த வேண்டும். / விசையை அழுத்துவதற்கு முன்பு, காட்டி இருந்த இடத்திலிருந்து தேடல் தொடங்கும். கோப்பின் இறுதி வரை தேடல் தொடரும். இது முன்னோக்கிய தேடல் ஆகும்.

வினாக்குறியும் (?) இதே பணியைத்தான் செய்யும். ஆனால் பின்னோக்கிய திசையில் தேடல் நடைபெறும். வினாக்குறியை அழுத்துவதற்கு முன்பு, காட்டி இருந்த இடத்தில் தொடங்கி, கோப்பின் தொடக்கம் வரை தேடல் தொடரும். இது பின்னோக்கிய தேடல் ஆகும்.

மாற்றம் செய்தல் (Changing)

cc கட்டளை ஒரு வரியை முற்றிலும் மாற்றியமைக்கப் பயன்படுகிறது. இக்கட்டளை முதலில் அவ்வரியை அழித்துவிட்டு, உள்ளீட்டுப் பாங்குக்கு மாறும். அந்த வரியில் புதிய விவரங்களை உள்ளிட்டு, Esc விசையை அழுத்த வேண்டும் இக்கட்டளை, dd மற்றும் o (ஒ) இரண்டும் சேர்ந்த கட்டளை ஆகும். r (replacement) கட்டளை, காட்டி இருக்கும் இடத்திலுள்ள ஒரேமுத்தை மாற்றியமைக்கப் பயன்படுகிறது. பிற கட்டளைகளைப் போல, இக்கட்டளை, உள்ளீட்டுப் பாங்குக்கு இட்டுச் செல்லாது. மாற்ற வேண்டிய எழுத்தைப் பதிந்தபின்னும், கட்டளைப் பாங்கிலேயேதான் இருப்பீர்கள். R (Replacement) கட்டளை, தொடர்ச்சியாக உரையை மேலேழுத (to overwrite) உதவும். இது, பிற உரைத்தொகுப்பிகளில் உள்ள மேலேழுதுதல் கட்டளைக்கு ஒப்பானது. ஆனால், உரையை மாற்றி எழுதும் போது நீங்கள் உள்ளீட்டுப் பாங்கினில் இருக்கிறீர்கள். எனவே உரையைப் பதிவு செய்தபின் Esc விசையை அழுத்தி, கட்டளைப் பாங்குக்கு மாறிக் கொள்ளவேண்டும். அந்த வகையில், இது பிற உரைத் தொகுப்பிகளிடமிருந்து வேறுபடுகிறது.

cc கட்டளை, ஒரு வரியை மாற்றியமைக்க உதவுவதுபோல, cw கட்டளை ஒரு சொல்லை மாற்றியமைக்க உதவுகிறது. dw கட்டளை ஒரு சொல்லை அழிக்கும்.

உரைத் தொகுப்பியில் இதுவரை நாம் அறிந்துகொண்ட கட்டளைகளைக் கீழேயுள்ள உரைப்பகுதியில் பயன்படுத்திப் பார்ப்போம்.

Having computer knowledge is an invaluable asset, it will do a world of good to young people seeking jobs.

- 1 Reliability:** Linux is a highly reliable system. Linux servers are not shut down for years together. Normally operating failures are unknown to Linux systems. It does not mean that you need not be vigilant. Do not forget the computer adage. If something can go wrong, it will.
- 2 Backward Compatibility:** Linux has excellent support for older hardware. It can run on different types of processors including the older ones. It can run the commands of its earlier version successfully.
- 3. Simple Upgrade and Installation:** The installation procedure of most Linux versions is menu driven and easy.
- 4. Suitable to any machine:** Suitable Linux version can run on any machine available now. This allows low investment for the hardware. The users, who have low configuration machines, prefer to use Linux OS compared to other OSs that require higher configurations.

மேற்கண்ட உரைப்பகுதியில் மூன்றாவது கருத்தை முதலாவதாக ஆக்கவேண்டும். 3 என்னும் எண்ணுக்கு முன்பாக, காட்டியைக் கொண்டு வரவும். கட்டளைப் பாங்கினில் இல்லாவிடில் Esc விசையை அழுத்திக் கொள்ளவேண்டும். இப்போது 2dd என்னும் கட்டளையைத் தருக. இரண்டு வரிகளும் அழிக்கப்பட்டு, இடையகத்தில் இருத்தப்படும். இப்போது இரண்டாவது வரிக்குச் செல்லுங்கள். இரண்டாவது வரிக்குச் செல்ல 2G கட்டளை தரவேண்டும். p (சிறிய எழுத்து) விசையை அழுத்துங்கள். உள்ளீட்டு பாங்குக்கு மாறுவீர்கள். Enter விசையை அழுத்தவும். மூன்றாவது கருத்து மூன்று, நான்காவது வரிகளாக இடம்பெறும். Esc விசையை அழுத்தி, கட்டளைப் பாங்குக்கு மாறுங்கள். ஆனால், பத்தி எண்களை மாற்றவேண்டியிருக்கும். எனவே, காட்டியை 3 என்ற எண்ணுக்கு அடியில் வைத்து, r (சிறிய எழுத்து) விசையை அழுத்தவும். எண் 3 அழிக்கப்பட்டு விடும். ஆனால், காட்டி மாற்று எழுத்துக்காகக் காத்திருக்கும். 1 என்ற எண்ணைப் பதியவும். நீங்கள் உள்ளீட்டுப் பாங்குக்கு மாறமாட்டீர்கள். மீண்டும், காட்டியை 1 என்ற எண்ணுக்குக் கொண்டுவந்து, 1-ஐ 2- ஆக மாற்றவும். இதேபோல 2 என்பதை 3 என மாற்றவும்.

இனி, "I want to stress the point that" என்னும் நிறைவடையாத தொடரை உரைப்பகுதியின் இறுதியில் சேர்க்க விரும்புகிறீர்கள். G விசையை அழுத்தவும். உரைப்பகுதியின் இறுதிக்குச் செல்வீர்கள். இப்போது மீண்டும் o (ஓ) விசையை அழுத்தவும். தோன்றுகின்ற வெற்று வரியில், I want to stress the point என உள்ளிடுங்கள். கோப்பின் தொடக்கத்துக்குச் செல்லுங்கள். 2yy கட்டளையைத் தரவும். முதலிரண்டு வரிகளும் இடையகத்தில் ஏற்றப்படும். மீண்டும் சொல்லுக்கு அடுத்து (ஓர் இடவெளி விட்டு) நிறுத்தவும். p விசையை அழுத்தவும். இப்போது, அந்த இரண்டு வரிகளும் that என்னும் சொல்லுக்கு அடுத்துப் பதிவாகும். இனி, Having என்ற சொல்லிலுள்ள H எழுத்தை h என மாற்ற வேண்டும். H என்ற எழுத்தின் கீழ், காட்டியை நிறுத்தவும். r விசையை அழுத்தவும். H எழுத்து அழிக்கப்பட்டு விடும். காட்டி, உங்கள் கட்டளைக்குக் காத்திருக்கும். h எழுத்தைப் பதியவும். இப்போது, உரைப்பகுதி கீழ்க்காணு மாறு மாறியிருக்கும்:

Having computer knowledge is an invaluable asset it will do a world of good to young people seeking jobs.

1 Simple Upgrade and Installation: The installation procedure of most Linux versions is menu driven and easy.

2 Reliability: Linux is a highly reliable system. Linux servers are not shut down for years together. Normally operating failures are unknown to Linux systems. It does not mean that you need not be vigilant. Do not forget the computer adage. If

something can go wrong, it will.

3 Backward Compatibility: Linux has excellent support for older hardware. It can run on different types of processors including the older ones. It can run the commands of its earlier version successfully.

4 Suitable to any machine: Suitable Linux version can run on any machine available now. This allows low investment for the hardware. The users, who have low configuration machines, prefer to use Linux OS compared to other OSs that require higher configurations.

3.10 செயல்தள உரைநிரல் (Shell Script)

'செயல்தள உரைநிரல்' என்பது லினக்ஸ் கட்டளைகள் அடங்கிய ஓர் உரைக்கோப்பு (text file) ஆகும். விஜ் தொகுப்பிபோன்ற அடிப்படையான உரைத் தொகுப்பி (Text Editor) ஏதேனும் ஒன்றில் அத்தகைய கோப்பினை நீங்கள் உருவாக்க முடியும். (எப்படி என்பதை இந்தப் பாடத்தின் இறுதியில் படிக்க உள்ளீர்கள்). குறிப்பிட்ட லினக்ஸ் கட்டளைகளின் தொகுதியை அதே வரிசையில் அடிக்கடி இயக்கவேண்டிய தேவை இருப்பின், செயல்தள உரைநிரலின் உதவியை நீங்கள் நாடலாம். கட்டளை வரியில் அத்தனை கட்டளைகளையும் வரிசையாக உள்ளிட்டு நிறைவேற்றுவது சலிப்பூட்டும் பணி மட்டுமன்று, வேண்டாத பணிச் சமையும் ஆகும். ஒன்று, நீங்கள் கட்டளைகளைப் பதியும்போது எழுத்துப் பிழை ஏற்படலாம் அல்லது கட்டளைகளின் வரிசையை மாற்றிவிடலாம். இரண்டு நிலையிலும் சிக்கலில் மாட்டிக் கொள்வீர்கள். இதுபோன்ற சூழ்நிலைகளில், செயல்தள உரைநிரல் கைகொடுக்கும். காரணம், கட்டளைகளை ஒரேயொரு முறைமட்டும் பிழையின்றிப் பதிவுசெய்து சேமித்துவைத்துக் கொண்டால் போதும். செயல்தள உரைநிரலில் உள்ளீட்டு/ வெளியீட்டுச் செயல்பாடுகளை நிறைவேற்ற முடியும். மாறிகளைக் (variable) கையாள முடியும்.

செயல்தள உரைநிரலை இயக்குதல்

லினக்ஸ் முறைமையில் உள்புகும் பொழுது, நீங்கள் பணியாற்றுவதற்குச் செயல்தளத்தின் சூழல் திறக்கப்படும். இச்செயல்தளம் 'உள்புகு செயல்தளம்' (Login Shell) எனப்படுகிறது. உங்களுக்குத் திறக்கப்படும் முன்னியல்பான செயல்தளம் (default shell) பாஷ் (Bash) செயல்தளம் ஆகும். பாஷ் செயல்தளம் நிரலாக்க மொழிகளைக் கையாளும் திறன் கொண்டது. இதன் உதவியோடு, சிக்கலான செயல்தள உரைநிரல்களை நீங்கள் உருவாக்க முடியும். ஒரு செயல்தள உரைநிரல், லினக்ஸ் கட்டளைகளை ஒருங்கிணைத்து, சிக்கல்களைத் தீர்த்துவைக்கப் பயன்படுகிறது.

சி-மொழியில் காணப்படும் கருவிகள் பலவற்றை லினக்ஸ் செயல் தளம் வழங்குகிறது. மாறிகளை உருவாக்கி மதிப்புகளை இருத்த முடியும். ஓர் உரைநிரல் கோப்பிலேயே மாறிகளை உருவாக்கலாம். நிரல் இயங்கும்போதே பயனர்கள் தரும் மதிப்புகளை மாறிகளில் இருத்தலாம். கட்டளைத் தூண்டுகுறியில், **sh** கட்டளையைத் தந்தால் ஒரு புதிய செயல்தளம் (**new shell**) உருவாக்கப்படும். இப்புதிய செயல்தளம், நடப்பு செயல்தளத்தின் உள்-செயல்தளம் அல்லது சேய்ச் செயல்தளம் என்று அழைக்கப்படும். செயல்தள உரைநிரலை நிறைவேற்ற, இந்தச் செயல் தளத்தைப் பயன்படுத்திக்கொள்ள முடியும். இந்த ஏற்பாடு பயனர்களின் அத்துமீறல்களுக்கு இடங்கொடுக்காமல் உள்புகு செயல்தளத்தைப் (**Login shell**) பாதுகாக்கிறது. ஏதேனும் விரும்பத்தகாத விளைவு ஏற்படின், சேய்ச் செயல்தளம்தான் பாதிக்கப்படும். அதுவும், உள்புகு செயல்தளத்துக்குத் தீங்கு ஏற்படாதவாறு, உடனடியாக அழிக்கப்பட்டுவிடும்.

செயல்தள உரைநிரலை மிகவும் கவனமாக உருவாக்க வேண்டும். அத்தகு கோப்பினை உருவாக்கும்போது, உங்களுக்குப் படிக்கும், எழு தும் சிறப்புரிமை இருக்கும். ஆனால், இயக்குகின்ற அனுமதி தானா கவே வழங்கப்பட மாட்டாது. இத்தகைய வரம்புக்குட்பட்ட செயலுரிமை களை வைத்துக்கொண்டே, செயல்தள உரைநிரலை நீங்கள் நிறைவேற்ற முடியும். கீழேயுள்ள கட்டளைகளுள் ஒன்றைப் பயன்படுத்துங்கள்:

\$ sh <filename>

அல்லது,

\$.<filename>

\$ தூண்டுகுறியில் நேரடியாக ஒரு செயல்தள உரைநிரலை இயக்க விரும்பினால், குறிப்பிட்ட நிரல்கோப்பின் கோப்பு அணுகு அனுமதியை (**File Access Permission - FAP**) மாற்றியமைக்க வேண்டும். கோப்பினை இயக் கும் (**execute**) அனுமதியை வழங்கவேண்டும். **chmod** கட்டளை மூலம் இதனைச் செய்துமுடிக்க முடியும். **edufile** என்னும் உரை நிரலை \$ குறியில் நேரடியாக இயக்க விரும்புகிறீர்கள்.

\$ chmod u+x edufile

\$ edufile

என்னும் கட்டளைகளைத் தாருங்கள்.

chmod கட்டளை **+x** உடன் சேர்ந்து, அனைத்துப் பயனருக்கும் இயக்கும் (**execute**) அனுமதியை வழங்கும். ஆனால் **u+x** கட்டளை, கோப் பின் உரிமையாளருக்கு மட்டும் இயக்கும் அனுமதியை வழங்குகிறது.

மேற்கண்ட உரைநிரலை இயக்கினால், நடப்புச் செயல்தளம் ஒரு புதிய செயல்தளத்தை உருவாக்கி அதில் உரைநிலை இயக்கும்.

3.11 மாறிகள் (Variables)

மாறிகள் என்பவை மதிப்புகளை இருத்திவைக்கும் நினைவக இருப்பிடங்களாகும். லினக்ஸ் மாறிகள் அனைத்தும் எழுத்துச் சரமாகவே கருதப்படும். இந்த மாறிகளைக் கொண்டு கணிதச் செயல்பாடுகள் எதையும் நிறைவேற்ற முடியாது என்பதுபோலத் தோன்றலாம். என்றாலும், `expr` மற்றும் `let` கட்டளைகள்மூலம் இந்த வரம்பை உடைத்தெறிய முடியும்.

மாறிகளை உருவாக்குதல் (Creating Variables)

ஏற்கெனவே குறிப்பிட்டபடி, பாஷ் செயல்தளமே உங்களின் முன்னியல்பான செயல்தளம் ஆகும். நீங்கலாக விரும்பி வேறொரு செயல்தளத்துக்கு மாறிக் கொள்ளாத வரையில் பாஷ் செயல்தளத்தில் மட்டுமே பணியாற்றமுடியும். ஒரு செயல்தளத்துக்குள் உருவாக்கப்படும் மாறிகள், 'செயல்தள மாறிகள்' என்று அழைக்கப்படுகின்றன. ஒரு பயனர் விரும்பும்போதெல்லாம், வெறுமனே மதிப்புகளை இருத்துவதன் மூலம் மாறிகளை உருவாக்கிக்கொள்ள முடியும். மதிப்பை இருத்தாமலே ஒரு மாறியை உருவாக்கவும் முடியும். மதிப்பிருத்தும் கட்டளையில் மதிப்பிருத்தும் செயற்குறி உட்பட வலப்பக்கத்தை அப்படியே விட்டுவிட வேண்டும்.

செயல்தள உரைநிரலில் உள்ள மாறிகளின் பெயர்கள் -

- ✱ அகரவரிசை எழுத்துகள், அடிக்கீறு (underscore), எண்களைக் கொண்டிருக்கலாம்.
- ✱ வியப்புக் குறி (!), உம் குறி (&), வெற்று இடவெளி ஆகியவை இடம்பெறக் கூடாது.
- ✱ முதல் எழுத்து, எண்ணாக இருக்கக் கூடாது.
- ✱ அதிக நீளம் கொண்டதாய் இருக்கக் கூடாது.
- ✱ கட்டளைச் சொற்கள், மாறிகளின் பெயராய் இருக்கக் கூடாது.
- ✱ ஏற்கத்தகு உரைநிரல் மாறிகளின் பெயர்கள்:

`file1, bookshell, book_shell, a+b, rs-paise`

- ✱ ஏற்கத் தகாத பெயர்கள்:

`a + b, a+ b, a!b, ab&, a=b`

மாறிகளை உருவாக்குவதற்கான கட்டளை அமைப்பு:

<variable_name> = <value>

குறிப்பு: ஒரு மாறியை அறிவிக்கும்போது, மதிருப்பித்தும் செயற்குறிக்கு (=) இருபுறமும் இடவெளி எதும் இருக்கக் கூடாது. இது, சி-மொழி நிரலாக்கத்தின் மதிப்பிருத்து கூற்றைப் போன்றது. = செயற்குறியின் இருபுறமும் வெற்று இடவெளி விட்டால், அது, சி-மொழி நிரலாக்கத்திலுள்ள == (நிகர்) செயற்குறிக்கு ஈடானதாகும்.

இருத்தப்படும் மதிப்பு (குறிப்பாகச் சர மதிப்பு), உட்பொதிந்த இடவெளிகள் போன்ற வரம்புக்குறிகளைக் (delimiters) கொண்டிருக்குமாயின், அந்த மதிப்பினை ஒற்றை அல்லது இரட்டை மேற்கோள் குறிகளுக்குள் தரவேண்டும். அதாவது, ஒற்றை மேற்கோள் குறி அல்லது இரட்டை மேற்கோள் குறி இரண்டையும் ஒன்றுக்கொன்று மாற்றாகப் பயன்படுத்தலாம் என்பதை அறிக. ஆனாலும் இரண்டுக்கும் இடையே மெல்லிய வேறுபாடு உண்டு. பின்னால் படிக்க உள்ளீர்கள்.

எடுத்துக்காட்டு:

name="Ezhil Kumaran"

மேற்கண்ட கட்டளையை இவ்வாறும் எழுதலாம்.

name='Ezhil Kumaran'

மதிப்புச் சரத்தில் இடவெளிகள் இல்லையெனில் மேற்கோள் குறிகள் கட்டாயமில்லை.

எடுத்துக்காட்டு:

name=llamathi

name='llamathi'

name="llamathi"

மேற்கண்ட அனைத்தும் ஒரே பொருளுடையவையே.

கீழ்க்காணும் மதிப்பிருத்தலை எடுத்துக் கொள்வோம்:

number=12

இந்த மதிப்பிருத்தலில் number என்னும் மாறியின் மதிப்பு எண்ணாக இருந்தபோதிலும், அது எண் மதிப்பு அல்ல. அது எழுத்துச் சரமே (character string) ஆகும். number என்னும் மாறியில் '1', '2' ஆகிய எழுத்துகளையே கொண்டுள்ளன. 12 (இரும் வடிவில் 1100) என்னும் எண்ணைக்

கொண்டிருக்கவில்லை. எனவே **number** என்னும் மாறியின் மீது இயல் கணிதத்தின் அடிப்படைச் செயல்பாடுகளை, அதாவது, கூட்டல், கழித்தல், பெருக்கல், வகுத்தல் ஆகிய செயல்பாடுகளை நிகழ்த்த முடியாது.

மாறிகளை எடுத்தாளல் (Referencing Variables)

ஒரு மாறியின் மதிப்பை எடுத்தாள **\$** குறியைப் பயன்படுத்த வேண்டும். **\$** குறி, அடுத்து இடம்பெறும் மாறியின் மதிப்பை எடுத்துத் தரும்.

எடுத்துக்காட்டு:

var1=\${var2},

var1, var2 ஆகியவை நினைவக இருப்பிடங்களைக் குறித்து நிற்கின்றன. **\${var2}** என்னும் கட்டளை, அந்த இருப்பிடத்தில் உள்ள மதிப்பை எடுத்துத் தருகிறது. பெறப்பட்ட மதிப்பின் நகல், **var1** மாறியில் இருத்தப்படுகிறது. அடைப்புக் குறிகள் கட்டாயமில்லை. ஆனால், ஒரு மாறியின் மதிப்போடு இன்னொரு மதிப்பை இணைக்க விரும்பினால் அடைப்புக் குறிகள் கட்டாயமாகும். எடுத்துக்காட்டாக, **father** என்னும் மாறியில் **John** என்னும் மதிப்பு உள்ளது, அதோடு **son** என்னும் மதிப்பை இணைத்து **son1** என்னும் மாறியில் இருத்த வேண்டுமெனில், கீழே உள்ளவாறு கட்டளை அமைக்க வேண்டும்:

\$ son1=\${father}son (முதல் **\$** குறி, தூண்டுகுறி)

son1 என்னும் மாறியில் **Johnson** என்னும் மதிப்பு இருக்கும்.

அடைப்புக் குறிகள் இல்லாவிட்டால் என்ன ஆகும்? விடை வெளிப்படையானது. அடைப்புக் குறிகள் இல்லாத கட்டளை இப்படி இருக்கும்:

\$ son1=\$fatherson

முதல் **\$**, தூண்டுகுறி, அடுத்த **\$**, மாறியைக் குறிப்பது.

father, son ஆகிய சொற்களுக்கிடையே வெற்று இடவெளி எதுவும் இல்லை என்பதால், **fatherson** என்பது மாறியின் பெயராக எடுத்துக்கொள்ளப்படும். அப்படியொரு மாறி இல்லை என்பதால், ஒரு பிழைகட்டும் செய்தி கிடைக்கும். அப்படியொரு மாறி இருந்துவிடும் எனில், அதன் மதிப்பு எவ்வித எச்சரிக்கைச் செய்தியுமின்றி **son1-**ல் இருத்தப்பட்டு விடும். விடை, விரும்பத்தகாதது ஆகிவிடும். (**father, son** இரண்டுக்கும் இடையே வெற்று இடவெளி விட்டால், மாறி புறக்கணிக்கப்பட்டுவிடும்).

மதிப்பைப் படித்து மாறியில் இருத்துதல் (Reading a value into a variable)

பயனரிடமிருந்து அவரது பெயரைப் பெறவேண்டுமெனில்,

“Please enter your name”

எனத் திரையில் காட்டவேண்டும். இதைப்பார்த்து, பயனர் தன்பெயரை உள்ளீடு செய்வார். அப்பெயரை நினைவகத்தில் இருத்திவைக்க ஏற் பாடு செய்யவேண்டும். இப்பணிகளை நிறைவேற்ற லினக்ஸில் இவ்வாறு கட்டளைகள் அமைக்க வேண்டும்:

echo “Please enter your name”

read name

echo கட்டளை ஒரு சரத்தைத் திரையில் காட்டும். இச்செய்தி, தன் பெயரை உள்ளீடு செய்யப் பயனரைத் தூண்டும். read கட்டளை செயல் படும்போது, name என்னும் மாறிக்குரிய மதிப்பை, பயனர் உள்ளிடும் வரை காத்திருக்கும். பயனர், தன் பெயரை உள்ளிட்டு Enter விசையை அழுத்தியதும், அப்பெயர் name என்னும் மாறியில் இருத்தப்படும். அடுத்து, நிரலில் வேறு கட்டளைகள் இருப்பின் அவை செயல்படுத்தப்படும். read கட்டளையை நேரடியாகச் செயல்தளத் தூண்டுகுறி (shell prompt) யில் தர முடியும். ஆனால், read கட்டளை பொதுவாக செயல்தள உரைநிரல்களில் லேயே பயன்படுத்தப்படுகிறது.

குறிப்பு: echo கட்டளையில் உரைச் சரத்திலுள்ள மேற்கோள் குறிகள் படிக்கத் தெளிவு கருதியே பயன்படுத்தப்பட்டுள்ளன.

3.12 கோவைகள் (Expressions)

expr மற்றும் let கட்டளைகள்

எவர் ஒருவரும், ஏதேனும் ஒரு காரணத்துக்காக, சிலவகைக் கணக் கீடுகளை, கணிப்புகளைச் செய்யாமல் நாளை நகர்த்திவிட முடியாது. எண்கணக்குக் கணிப்பீடுகள் இல்லாத ஓர் உலகை நினைத்துப் பார்க்க முடியுமா? விடை நிச்சயமாக “முடியாது” என்பதுதான். பெரும்பாலான செயல்தளங்கள் எண்வகை மாறிகளை ஏற்பதில்லை. அனைத்து மாறி களும் எழுத்துச் சரமாகவே கருதப்படுகின்றன. என்றாலும் செயல்தளத் தில் நிரல் எழுதும்போது, மாறிகளைக் கணித முறையில் கையாள வழி யிருக்க வேண்டும். இது தவிர்க்கமுடியாத தேவையாகும். expr மற்றும் let கட்டளைகள் மூலம் இது சாத்தியம். expr கட்டளை, கணக்கீட்டுக் கோவைகளின் விடையைக் கணிக்கப் பயன்படுகிறது. இக்கட்டளையின் வெளியீடு அடிப்படை வெளியீட்டுக்கு (திரைக்கு) அனுப்பப்படுகிறது.

எடுத்துக்காட்டு:

\$ expr 21 + 51

என்னும் கட்டளை 72 என்னும் விடையைத் திரையில் காட்டும். + குறியின் இருபுறமும் ஓர் இடவெளி (Space) உள்ளது என்பதைக் கவனிக்கவும். expr கட்டளையில் மாறிகளைக் கையாளும் முறையைக் காண்போம்.

\$num1=7

\$num2=3

\$expr \$num1 + \$num2

இதன் வெளியீடு திரைக்கு அனுப்பப்படும் என்பதால், திரையில் 10 எனக் காட்டப்படும். \$ குறி பயனருக்கான தூண்டுக்குறி (prompt) ஆக இல்லாதபோது, ஒரு மாறியின் மதிப்பைச் சட்டுவதற்குப் பயன்படுகிறது. எனவே \$num1 என்ற இடத்தில் 7 எனவும், \$num2 என்னும் இடத்தில் 3 எனவும் எடுத்துக் கொள்ளப்படும். அடுத்து, expr கட்டளை இந்த எழுத்துகளை எண்களாக மாற்றும். அதன்பிறகே அவற்றின் கூட்டுத் தொகையைக் கண்டறியும்.

expr கட்டளை +, -, *, / ஆகிய கணக்கீடுகளைச் செய்யும். ஆனால், * செயற்குறியைப் பயன்படுத்தும்போது மிகவும் கவனமாக இருக்க வேண்டும். ஏனெனில், * குறி, பதிலீட்டுக் குறியீடாகவும் பயன்படுத்தப்படுகிறது. எனவே, பெருக்கல் கணக்கில் அது, வேறுபடுத்திக் காட்டப்பட வேண்டும். * என்று எழுதினால், பெருக்கல் குறியாகக் கொள்ளப்படும்.

\$ expr 1 / 2

என்னும் கட்டளை 0 என விடைதரும். 0.5 என்ற விடை காட்டாது. / குறிக்கு முன்னும் பின்னும் வெற்று இடவெளி உள்ளதைக் கவனிக்கவும்.

\$expr 0.5 / 2

எனக் கட்டளை தந்தால், என்ன கிடைக்கும்? ஒரு பிழைகட்டும் செய்தியே விடையாகக் கிடைக்கும். ஏனெனில், பதின்மப் புள்ளி (decimal point), வெறும் புள்ளி (dot) யாகத்தான் கருதப்படும். 0.5 என்பது ஓர் எண்ணாகவே ஏற்றுக் கொள்ளப்படமாட்டாது.

let கட்டளை, கணக்குகளின் விடைகாணவும், இரண்டு மதிப்புகளை ஒப்பிடவும் உதவுகிறது. அதன் கட்டளை வடிவம்:

\$ let < மதிப்பு1 > < செயற்குறி > < மதிப்பு2 >

இங்கே செயற்குறி என்பது கணக்கீட்டுச் செயற்குறியாக (arithmetic

operator) இருக்கலாம். அல்லது ஒப்பீட்டுச் செயற்குறியாக (relational operator) இருக்கலாம். **let** கட்டளை **expr** கட்டளையைவிட மேம்பட்டது. **let** கட்டளை எந்த மாறியையும் மதிப்பிட்டு, அதன் மதிப்பை எண்கணக்கு மாறியாக மாற்றிவிடும். இச்செயல்திறன், செயல்தள உரைநிரலில் கட்டுப்பாட்டுக் கட்டமைப்புகளில் பயன்படுத்திக் கொள்ளப்படுகிறது. **expr** கட்டளையில் செயற்குறிக்கு முன்னும் பின்னும் இடவெளி (space) தேவை. ஆனால் **let** கட்டளையில் செயற்குறிக்கு முன்னதாகவோ, அடுத்தோ இடவெளி/இடவெளிகள் தேவை இல்லை. அவ்வாறு இடவெளி/இடவெளிகள் விட விரும்பினால் விடலாம். ஆனால் கணிப்புக் கோவை முழுவதையும் மேற்கோள் குறிகளுக்குள் தரவேண்டும். **let** கட்டளையில் குறிப்பிடும் ஒரு கணப்பின் விடையை மாறியில் இருத்தவில்லை எனில், விடை திரையில் காட்டப்படும்.

எடுத்துக்காட்டு:

```
$ let pr = 5*10
```

```
$ echo "The product is $pr"
```

```
The product is 50
```

குறிப்பு: 1) பெருக்கல் குறியை * எனக் குறிப்பிட வேண்டிய தில்லை. 2) **let** கட்டளையின் கணிப்புக் கோவையில் மதிப்பிருந்து செயற்குறிக்கு (=) இடப்பக்கம் கட்டாயமாக ஒரு மாறியின் பெயர் இடம்பெற வேண்டும். **expr** கட்டளையில் தேவையில்லை.

செயற்குறியின் முன்னும் பின்னும் இடவெளி/இடவெளிகள் விட விரும்பினால், முழுக் கணக்கீட்டையும் மேற்கோள் குறிக்குளுக்குள் அமைக்கவேண்டும்.

எடுத்துக்காட்டு:

```
$ let "pr = 5 * 10"
```

```
$ echo "The product is $pr"
```

```
The product is 50
```

விடையை மாறியில் இருத்தினால் விடை திரையில் காட்டப்பட மாட்டாது. விடையைத் திரையில் காட்டவேண்டுமெனில் **echo** கட்டளையைப் பயன்படுத்த வேண்டும்.

எடுத்துக்காட்டு:

\$ let "sum = 2 + 4"

\$ echo "The sum is \$sum"

The sum is 6

உரைநிரலாக்கத்தில் let கட்டளையைப் பயன்படுத்திக் கீழே உள்ள வாறு மதிப்பு இருத்தலும் சாத்தியமே.

let a = 0

let a = a+1

குறிப்பு: 2.3, 0.5 போன்ற பதின்ம எண்களை let கட்டளையும் புரிந்துகொள்ளாது.

ஒற்றை மேற்கோள் குறிக்கும் இரட்டை மேற்கோள் குறிக்கும் இடையே மெல்லிய வேறுபாடு உள்ளதெனக் கூறினோம். இனி, அந்த வேறுபாட்டைப் பார்ப்போம். name என்னும் மாறியில் llamathi என்னும் பெயர் இருப்பதாகக் கொள்வோம்.

echo "The given name is \$name"

echo 'The given name is \$name'

என்னும் கட்டளைகளைத் தருகிறீர்கள். இவற்றின் விடை இவ்வாறு இருக்கும்:

The given name is llamathi

The given name is \$name

இரட்டை அடைப்புக் குறிகளுக்குள் \$ குறி மாறியின் மதிப்பைச் சுட்டுவதாகக் கொள்ளப் படுகிறது. எனவே அந்த இடத்தில் llamathi என்னும் பெயர் பதிலிடப்படுகிறது. ஆனால் ஒற்றை மேற்கோள் குறிகள், உள்ளிருக்கும் செய்தியை அப்படியே வெளியிட்டுவிடும். அதாவது, \$name என்பது அப்படியே இடம்பெறும். மாறியின் மதிப்பைச் சுட்டுவதாய்க் கொள்ளப்படாது.

3.13 கட்டளைப் பதிலீடு (Command Substitute)

ஒரு செய்தியின் நடுவில் கணிப்பொறியின் நடப்புத் தேதியைக் காட்ட விரும்புகிறீர்கள். \$date என்னும் கட்டளை எந்தப் பலனையும் தராது. (\$ குறி, மாறியிலிருக்கும் மதிப்பினை மட்டுமே எடுத்துத் தரும். கட்டளை தரும் மதிப்புகளைப் பெற்றுத் தராது). ஒரு கட்டளையிலிருந்து

விடையப் பெற்றுத் தரவேண்டுமெனில் அக்கட்டளையைப் பின்சாய்வு மேற்கோள் குறிக்களுக்குள் (எடுப்பொலிக் குறி `) அமைக்க வேண்டும். ஒரு செய்தியின் அங்கமாக நடப்புத் தேதியைக் காட்ட, **date** கட்டளையைப் பின்சாய்வு மேற்கோள் குறிக்களுக்குள் (எடுப்பொலிக் குறி) தர வேண்டும். இக்குறி, விசைப்பலகையில் எண் 1 (அல்லது!) விசைக்கு முன்னதாக இருக்கும். நெளிவுக் குறியும் (~ tilde), பின்சாய்வு மேற்கோள் குறியும் (`backward quote) ஒரே விசையில் மேலும் கீழும் இருக்கும். நடப்புத் தேதியைக் காட்ட இவ்வாறு கட்டளை அமைக்க வேண்டும்.

எடுத்துக்காட்டு:

\$ echo "Today's date is `date +% D` "

செய்தி திரையில் காட்டப்படும்போது, `date + % D` என்றிருக்கும் இடத்தில் கணிப்பொறியின் தேதி இடம்பெறும். **date** கட்டளைக்கும் + குறிக்கும் இடையே ஓர் இடவெளி இருப்பதைக் கவனியுங்கள். அது இல்லையேல் இக்கட்டளை செயல்படாது. (+%D குறியீடு mm/dd/yy வடிவில் தேதியை மட்டும் பிரித்துத் தரும்). செயல்தளம், முதலில் கட்டளையின் வெளியீட்டைச் செய்தியில் இடம்பெறச் செய்யும். பிறகு **echo** கட்டளையை நிறைவேற்றும். இத்தகைய 'கட்டளைப் பதிலீடு' (**comma substitution**) மூலம், ஒரு கட்டளையின் வெளியீட்டை ஒரு மாறியில் இருத்தவும் முடியும்.

எடுத்துக்காட்டு:

cfiles = `ls *.c | wc - l`

இக்கட்டளை, நடப்புக் கோப்பகத்தில் .c என முடியும் கோப்புகளின் எண்ணிக்கையை **cfiles** என்னும் மாறியில் இருத்தும்.

ஏற்கெனவே சொல்லியபடி, **expr** கட்டளையின் வெளியீடு, அடிப்படை வெளியீட்டுக்குச் செல்லும். அதற்குப் பதிலாக, வெளியீட்டை ஒரு மாறியில் பதிவுசெய்ய வேண்டியிருந்தால், கட்டளைப் பதிலீட்டைப் பயன்படுத்தலாம். எடுத்துக்காட்டாக,

\$ var1=5

\$ var1=`expr \$var1 + 20`

என்னும் கட்டளை **var1** என்னும் மாறியில் 25 என்னும் மதிப்பை இருத்தும்.

expr கட்டளை ஒற்றைப் பின்சாய்வு மேற்கோள் குறிக்களுக்குள் உள்ளது. செயல்தளம் முதலில் கட்டளை இருக்கும் இடத்தில் அதன் வெளி

யீட்டை இட்டு நிரப்பும். பிறகு, மீதிக் கட்டளையைச் செயல்படுத்தும்.

பிற உயர்நிலை மொழிகளில் உள்ள if, for, while போன்ற கட்டளை களை உயர்நிலை செயல்தள உரைநிரலில் (Advanced Shell Script) பயன்ப டுத்த முடியும்.

3.14 லினக்ஸின் சிறப்புக் கூறுகள்

நம்பகத் தன்மை (Reliability)

லினக்ஸ், மிகவும் நம்பகத் தன்மை வாய்ந்த இயக்க முறைமை ஆகும். லினக்ஸ் சேவையகங்கள் (servers) ஆண்டுக் கணக்கில் நிறுத் தப்படாமல் தொடர்ந்து செயல்படுகின்றன. பெரும்பாலும் லினக்ஸ் முறை மைகளில் செயல்பாட்டு நிலைகுலைவு நிகழ்வதே கிடையாது. நீங்கள் எச்சரிக்கையுடன் இருக்க வேண்டியதில்லை என்பது இதற்குப் பொரு ளாகாது. கணிப்பொறி முதுமொழியை மறந்துவிட வேண்டாம். “எதுவும் பிழையாகலாம், இதுவும் பழுதாகலாம்”.

பின்னோக்கிய ஒத்திசைவு (Backward Compatibility)

பழைய கணிப்பொறிகளிலும் லினக்ஸ் நன்கு செயல்படும். பழைய செயலிகளிலும் (processors) செயல்படக்கூடியது. பல வகையான செய லிகளிலும் செயல்படவல்லது. முந்தைய பதிப்புகளில் உள்ள லினக்ஸ் கட்டளைகளை, புதிய பதிப்புகளிலும் வெற்றிகரமாக இயக்க முடிகிறது.

எளிய மேம்படுத்தலும் நிறுவுகையும் (Simple Upgrade and Installation)

பெரும்பாலான லினக்ஸ் பதிப்புகளின் நிறுவுகைச் செயல்முறை பட்டி (Menu) அடிப்படையிலானது, மிகவும் எளிதானது.

எந்தக் கணிப்பொறிக்கும் ஏற்றது (Suitable to any machine)

இப்போதிருக்கும் எந்தக் கணிப்பொறியிலும் அதற்குரிய லினக்ஸ் பதிப்பை இயக்க முடியும். எனவே வன்பொருளுக்கென அதிகமான முதலீடுகள் செய்ய வேண்டியதில்லை. குறைந்த திறனுள்ள கணிப்பொறி வைத்திருப்பவர்கள், அதிகத் திறனுள்ள வன்பொருள் அமைப்பு தேவைப் படுகின்ற, பிற இயக்க முறைமைகளைக் காட்டிலும் லினக்ஸையே பெரிதும் விரும்புகின்றனர்.

ஜியூஐ கிடைமுகம் (GUI Interface)

கேடிஐ, ஜினோம், சேவையகம் ஆகிய வரைகலை இடைமுகங்கள் லினக்ஸில் உள்ளன. லினக்ஸ் முறைமையை இரண்டு உட்பிரிவுகளாகப் பிரிக்கலாம். கேடிஐ, ஜினோம் இரண்டும் ஏறத்தாழ விண்டோஸ் 98

முறைமை வழங்கும் அனைத்து வசதிகளையும் வழங்குகின்றன.

பற்பல வெளியீட்டாளர்கள் (Multiple Distributors)

லினக்ஸ் மென்பொருளை பல நிறுவனங்கள் வெளியிடுகின்றன. ஒவ்வொன்றும் அதற்கே உரிய கூடுதல் வசதிகளைக் கொண்டுள்ளன. இதன் காரணமாக, லினக்ஸ் விற்பனைச் சந்தை வாங்குவோர் சந்தை (Buyers Market) யாக உள்ளது. ரெட்ஹேட் (Redhat), கால்டெரா (Caldera), மாண்ட்ரேக் (Mandrake), டெபியன் (Debian), ஸ்லேக்வேர் (Slackware) ஆகியவை லினக்ஸ் வெளியீடுகளுள் சில.

நச்சுநிரல் தாக்குதல் இல்லை (No Virus Attack)

கணிப்பொறித் தொழில்நுட்பத்தில் மிகவும் அச்சமூட்டும் சொல் 'நச்சு நிரல்'. நச்சுநிரல் உண்மையில் இணையத்தின் பரவலை மட்டுப்படுத்துகிறது. பொதுவாக லினக்ஸ் நச்சுநிரல் தாக்குதலில்லை என்று கூறப்படுகிறது. இப்போதெல்லாம் லினக்ஸ் தாக்கும் நச்சுநிரல்களும் வந்து விட்டன.

பாதுகாப்பு வசதிகள் (Security Features)

இணையக் குறும்பினில் நாட்டமுள்ள சிலர் மற்றவர்களது பணியில் பேரழிவை விளைவிக்கின்றனர். இத்தகைய தாக்குதல்களுக்கு எதிராக லினக்ஸ் அருமையான பாதுகாப்பு வசதிகளைக் கொண்டுள்ளது. இதன் காரணமாகவே இணையச் சேவை வழங்குவோர் (Internet Service Providers - ISPs) பலரும் லினக்ஸ் முறைமைக்கு மாறுகின்றனர்.

ஏராளமான பயனர்கள் பணியாற்றலாம் (Can Support a High User Load)

லினக்ஸில் ஒரே நேரத்தில் ஏராளமான பயனர்கள் பணியாற்ற முடியும்.

உருவாக்க நூலகங்கள் (Development Libraries)

சி++, பேரல் போன்ற பல உருவாக்க மொழிகளுக்கான சிறந்த பணித்தளத்தை (platform) லினக்ஸ் வழங்குகிறது.

தொகுப்புரை

- ◆ (nyy அல்லது nY), p கட்டளைகள் n வரிகளை நகலெடுக்கும்
- ◆ லினக்ஸில் எந்த உரைத் தொகுப்பியைப் பயன்படுத்தியும் செயல் தள உரைநிரல்களை உருவாக்கலாம்.
- ◆ கணக்கீட்டுக் கோவைகளின் விடையைக் கணிக்க expr கட்டளை பயன்படுகிறது.
- ◆ let கட்டளை expr கட்டளையைவிட மேலானது.
- ◆ vi கட்டளை விஐ தொகுப்பியை இயக்குகிறது.
- ◆ விஐ தொகுப்பி இருவகைப் பாங்கினில் செயல்படும். உள்ளீட்டுப் பாங்கு மற்றும் கட்டளைப் பாங்கு. தற்போது பயனர் எந்த வகைப் பாங்கினில் இருக்கிறார் என்பதை அறிய Esc விசையைப் பயன்படுத்தலாம்.
- ◆ மேற்கண்ட இரு பாங்குகள் தவிர, வரிப் பாங்கு என்கிற மூன்றாவது பாங்கினையும் அறிமுகப்படுத்தியுள்ளோம்.
- ◆ ஒன்றுக்கு மேற்பட்ட கட்டளைகளை ஒரே கட்டளையாக இயக்க, 'கட்டளைப் பதிலீடு' (Command Substitution) பயன்படுகிறது.

பயிற்சி வினாக்கள்

புள்ளியிட்ட இடங்களை நிரப்புக

1. \$vi filename. இரண்டு பெரிய எழுத்து _____ பயன்படுத்தி, கோப்பினைச் சேமிக்கலாம்.
2. :w என்ற கட்டளை, கோப்பினை _____, _____ .
3. Esc விசையை அழுத்தும்போது, 'பீப்' ஒலி கேட்டால், நீங்கள் _____ பாங்கினில் உள்ளீர்கள் என்று பொருள்.
4. ndd மற்றும் _____ கட்டளைகள் குறிப்பிட்ட உரைப்பகுதியை விரும்பும் இடத்துக்கு நகர்த்தும்.
5. _____ கட்டளை எண்கணக்குக் கணிப்புகளை மிகவும் திறனுடன் செய்து முடிக்கிறது.
6. செயல்தள உரைநிரலில் _____ கட்டளை, செயல்பாட்டை நிறுத்திவிடும்.