Termux Commands: Beginner to Advanced

Termux is a powerful terminal emulator for Android that provides access to a Linux command line environment. It allows you to install a variety of Linux packages using the package manager pkg or apt. Here's a guide that spans from beginner to advanced Termux commands, designed to help you get acquainted with its capabilities:

Basic System Operations

- `pkg update` - Update package list.
- `pkg upgrade` - Upgrade installed packages.
- `pkg install <package>` - Install a specific package.
- `pkg uninstall <package>` - Uninstall a specific package.
- `pkg list-installed` - List all installed packages.
- `pkg search <query>` - Search for a package.
- `termux-setup-storage` - Setup access to device storage.
- `exit` or `logout` - Exit Termux session.
- `clear` - Clear the terminal screen.

File and Directory Management

- `ls` - List directory contents.
- `cd <directory>` - Change the current directory.
- `pwd` - Print the current directory.
- `mkdir <directory>` - Create a new directory.
- `rmdir <directory>` - Remove an empty directory.
- `rm <file>` - Remove a file.
- `rm -r <directory>` - Remove a directory and its contents.
- `cp <source> <destination>` - Copy files or directories.
- `mv <source> <destination>` - Move files or directories.
- `touch <file>` - Create a new file.
- `cat <file>` - Display file content.
- `less <file>` - View file content page by page.
- `nano <file>` - Edit file using Nano.
- `vim <file>` - Edit file using Vim.

Networking and Internet

- `ping <hostname>` - Check the network connection to a host.

- `wget <URL>` - Download files from the internet.
- `curl <URL>` - Transfer data from or to a server.
- `ssh <user>@<host>` - Connect to a remote server via SSH.
- `scp <source> <destination>` - Securely copy files between hosts.
- `ftp <hostname>` - Connect to an FTP server.
- `nslookup <domain>` - Query DNS lookup information.
- `whois <domain>` - Retrieve domain registration information.

System Information and Monitoring

- `top` - Display ongoing system processes.
- `htop` - An interactive process viewer (requires installation).
- `df` - Display disk space usage.
- `du` - Estimate file space usage.
- `free` - Display memory usage.
- `uptime` - Show how long the system has been running.
- `uname -a` - Show system information.
- `netstat` - Display network connections.
- `ifconfig` - Display or configure network interfaces.

Package Management

- `apt list` - List packages available for installation.
- `apt show <package>` - Display package information.
- `apt clean` - Clean up downloaded package files.
- `dpkg -l` - List all installed Debian packages.

Text Processing

- `grep <pattern> <file>` - Search for a pattern in a file.
- `sed 's/<find>/<replace>/' <file>` - Find and replace text within a file.
- `awk '{print $1}' <file>` - Process and analyze text files.
- `sort <file>` - Sort lines of text files.
- `uniq <file>` - Report or omit repeated lines.
- `cut -d':' -f1 <file>` - Remove sections from each line of files.

Scripting and Programming

- `python <script.py>` - Execute a Python script.
- `perl <script.pl>` - Execute a Perl script.

- `bash <script.sh>` - Execute a Bash script.
- `gcc <source.c> -o <output>` - Compile a C program.
- `java <MainClass>` - Execute a Java program.

Version Control

- `git clone <repository>` - Clone a git repository.
- `git pull` - Update local repository to the newest commit.
- `git push` - Update remote repository with local changes.
- `git status` - Show the working tree status.
- `git commit -m "message"` - Commit changes to the repository.

Miscellaneous Commands

- `tar -xzf <file.tar.gz>` - Extract a tar.gz file.
- `zip -r <archive.zip> <directory>` - Create a zip archive of a directory.
- `unzip <archive.zip>` - Extract a zip archive.
- `echo <text>` - Display a line of text.
- `date` - Display or set the system date and time.
- `cal` - Display a calendar.
- `bc` - An arbitrary precision calculator language.

Advanced Usage

- `ssh-keygen` - Generate an SSH key pair.
- `ssh-copy-id <user>@<host>` - Copy SSH key to a remote host.
- `screen` - Multiplex terminal, run multiple sessions inside one terminal.
- `tmux` - Terminal multiplexer, similar to screen but more powerful.
- `find . -type f -name "*.txt"` - Find files by name.
- `chmod +x <script.sh>` - Make a script executable.
- `crontab -e` - Edit crontab entries, schedule tasks.
- `iptables -L` - List iptables firewall rules.
- `nmap <host>` - Network exploration tool and security / port scanner.
- `python -m SimpleHTTPServer` - Start a simple HTTP server (Python 2).
- `python3 -m http.server` - Start a simple HTTP server (Python 3).
- `ffmpeg -i input.mp4 output.mp3` - Convert video files to audio.
- `termux-api` - Access Termux API functions (requires Termux:API app).

Beginner Commands

1. **Update and Upgrade Packages**

```
pkg update && pkg upgrade
```

This command updates the list of available packages and their versions, then upgrades the installed packages to their latest versions.

2. **Install a Package**

```
pkg install [package_name]
```

Replace `[package_name]` with the name of the package you want to install. For example, `pkg install python` to install Python.

3. **List Installed Packages**

```
pkg list-installed
```

Displays a list of all packages that are currently installed.

4. **Uninstall a Package**

```
pkg uninstall [package_name]
```

Removes the specified package from your system.

5. **Search for a Package**

```
pkg search [keyword]
```

Searches the repositories for a package with the specified keyword.

Intermediate Commands

6. **Access Storage**
   First, grant Termux access to device storage:

```
termux-setup-storage
```

After allowing storage permissions, you can navigate to your storage using `cd` command.

7. **SSH into Another Machine**

```
pkg install openssh
 ssh user@hostname
```

Replace `user` with your username and `hostname` with the IP address or domain of the remote server.

8. **Using Git**

```
pkg install git
 git clone [repository_url]
```

Clone a repository from GitHub or any other Git repository.

9. **Running Python Scripts**

```
pkg install python
 python script.py
```

Runs a Python script named `script.py`.

10. **Wget to Download Files**
    `bash pkg install wget wget [file_url]`
    Downloads files from the internet.

Advanced Commands

11. **Use Vim or Nano as an Editor** `pkg install vim` or `pkg install nano` Install and use Vim or Nano text editors for editing files.
12. **Cron Jobs**
    Use `crontab` to manage cron jobs, scheduling scripts or commands to run at specified times.
13. **Custom Scripts**
    Write custom shell scripts to automate tasks. Make sure to give them execute permissions: `chmod +x script.sh`
14. **Networking Tools**
    Install networking tools like `nmap` for network exploration and security auditing: `pkg install nmap`
15. **Python Virtual Environments** `pkg install python python -m venv myenv source myenv/bin/activate` Create and activate Python virtual environments for project-specific dependencies.
16. **Package Creation**
    Advanced users can create their own packages for Termux by following the packaging guide provided in the Termux GitHub repositories.
17. **SSH Server** `pkg install openssh sshd` Set up an SSH server on your device to allow remote access.
18. **Termux API** `pkg install termux-api` Access device features such as the camera, GPS, and SMS through command line scripts.

19. **Encrypt Files with GPG** `pkg install gnupg gpg -c file.txt` Encrypt files using GnuPG.
20. **Web Development**
    Install web development tools like `nodejs`, `php`, or `ruby` and use Termux as a portable web development environment.

This guide covers a broad spectrum of commands and uses, but the possibilities with Termux are vast. You can explore further by trying out different packages, reading the documentation, and participating in the Termux community.