Linux Shells, Profiles, Umask & Crontab — Practical README

This README is a **hands-on guide** to understand: 1. Shell types (login, non-login, interactive, non-interactive) 2. Startup files (`/etc/profile`, `~/.profile`, `~/.bash_profile`, `~/.bashrc`) 3. `umask` (what, why, where) 4. `crontab` (how to schedule jobs correctly)

1️⃣ Shell Types (with REAL examples)

What is a shell?

A shell is a program that reads commands and executes them (e.g., `bash`, `sh`).

You can check your shell:

```
echo $SHELL
ps -p $$
```

A. Login Shell

A shell started **as part of logging in**.

Examples

```
ssh user@server
sudo -i
su -
bash --login
```

Behavior - Reads `/etc/profile` - Reads ONE of: `~/.bash_profile` → `~/.bash_login` → `~/.profile`

B. Non-login Shell

A shell started **inside an existing session**.

Examples

```
bash
sh
```

Behavior - Does NOT read `/etc/profile` - Reads `~/.bashrc` (if interactive)

C. Interactive Shell

You can type commands at a prompt.

Examples

```
bash
ssh user@server
```

Check:

```
echo $-   # contains 'i' if interactive
```

Behavior - Reads `~/.bashrc`

D. Non-interactive Shell

Runs commands automatically (scripts, automation).

Examples

```
./script.sh
bash script.sh
bash -c "ls"
cron jobs
```

Behavior - Reads NO startup files (unless forced)

⚫Mental Rule

```
LOGIN?        → profile files
INTERACTIVE?  → bashrc
```

2️⃣Shell Startup Files (Who runs what?)

`/etc/profile` (System-wide, LOGIN only)

- Applies to **all users**
- Runs **only for login shells**

Use for - Default `PATH` - Default `umask` - Global environment variables

Example

```
# /etc/profile
export PATH="$PATH:/opt/tools"
umask 0022
```

`~/.profile` (User, GENERIC login)

- POSIX / shell-agnostic
- Used by `sh`, `dash`, and by `bash` **only if** `.bash_profile` is missing

Use for - Portable login settings

Example

```
# ~/.profile
umask 0022
export PATH="$PATH:$HOME/bin"
```

`~/.bash_profile` (User, BASH login)

- Bash-specific login file
- Preferred by `bash` over `.profile`

Best practice

```
# ~/.bash_profile
umask 0022
source ~/.bashrc
```

`~/.bashrc` (INTERACTIVE only)

- Runs every time an interactive terminal opens

Use for - Aliases - Prompt - Functions - Dev-only exports

Example

```
# ~/.bashrc
alias ll='ls -ltr'
export EDITOR=vim
```

❌**Do NOT put** `umask` **here in production**

Bash Login Order (important)

```
/etc/profile
~/.bash_profile  (if exists, stop)
~/.bash_login    (if .bash_profile missing)
~/.profile       (if both missing)
```

Only **ONE** user file is executed.

3️⃣UMASK (Very important)

What is umask?

`umask` defines **default permissions** for newly created files/directories.

Defaults

- Files start from `666`
- Directories start from `777`

Final permission:

```
final = base - umask
```

Examples

```
umask 0022
```

- File: `666 - 022 = 644` - Dir: `777 - 022 = 755`

```
umask 0077
```

- File: `600` - Dir: `700`

Where to set umask?

Scope	File
System-wide	`/etc/profile`
User login	`~/.bash_profile` or `~/.profile`
❌Avoid	`~/.bashrc`

Key rule

> `umask` is **per process** and applied at **login time**.

To apply changes → **re-login** (no reload command).

4️⃣CRONTAB (Job Scheduling)

What is cron?

- `cron` = background daemon
- `crontab` = schedule table

Cron jobs are: - Non-interactive - Non-login - Minimal environment

---

Crontab Format

```
* * * * * command
| | | | |
| | | | └── Day of week (0–7 Sun)
| | | └──── Month (1–12)
| | └────── Day of month (1–31)
| └──────── Hour (0–23)
└────────── Minute (0–59)
```

User Crontab

```
crontab -e          # edit
crontab -l          # list
crontab -e -u user  # other user
```

Example

```
*/5 * * * * /bin/bash /home/app/health.sh >> /home/app/health.log 2>&1
```

System-wide Crontab

File: `/etc/crontab`

Format includes USER

```
0 2 * * * root /bin/bash /scripts/backup.sh >> /var/log/backup.log 2>&1
```

Cron Directories (run-parts)

Directory	Runs
`/etc/cron.hourly/`	Hourly
`/etc/cron.daily/`	Daily
`/etc/cron.weekly/`	Weekly
`/etc/cron.monthly/`	Monthly

Drop executable scripts here.

Special Time Strings

```
@reboot   /scripts/startup.sh
@daily    /scripts/cleanup.sh
@weekly   /scripts/report.sh
```

Cron Gotchas (must remember)

- Always use **full paths**
- Cron does NOT read `.bashrc`
- Set PATH manually if needed
- Always log output

Final Mental Map

```
LOGIN  → /etc/profile, ~/.bash_profile, ~/.profile
INTERACTIVE → ~/.bashrc
SCRIPTS / CRON → no startup files
UMASK → set at login
CRON → minimal env, full paths
```

Interview One-liner

> “Linux startup behavior depends on login vs interactive shells; profile files apply at login, bashrc applies to interactive shells, umask is process-specific, and cron runs non-interactively with a minimal environment.”

✅End of README