



# Linux Service and Process Management



## What Are Services (Daemons)?

- **Services (daemons)** run in the background and perform critical system tasks.
- Named with a trailing **d** (e.g., **sshd**, **systemd**).

### Types

1. **System Services** – Essential for system startup (e.g., **init**, **systemd**).
2. **User-Installed Services** – Add-ons like web servers (e.g., **Apache**).



## Common Service Goals

- Start/Restart
- Stop
- Status check
- Enable/Disable on boot
- Locate services



## systemctl Commands

Command	Description
<code>systemctl start ssh</code>	Start SSH service
<code>systemctl status ssh</code>	Check service status
<code>systemctl enable ssh</code>	Enable service on boot
<code>systemctl list-units --type=service</code>	List active services
<code>journalctl -u ssh.service</code>	View service logs



## Viewing Processes

```
ps aux | grep <process>
```

### Example

```
ps aux | grep ssh
```



## Killing Processes

Command	Description
<code>kill &lt;PID&gt;</code>	Send default SIGTERM
<code>kill -9 &lt;PID&gt;</code>	Force kill (SIGKILL)
<code>kill -l</code>	List all signals
<code>pkill &lt;name&gt;</code>	Kill by process name
<code>killall &lt;name&gt;</code>	Kill all with name

### Most used signals

- `1` – SIGHUP
- `2` – SIGINT (Ctrl+C)
- `9` – SIGKILL
- `15` – SIGTERM
- `20` – SIGTSTP (Ctrl+Z)



## Background/Foreground Process Management

### Suspend a process

`[Ctrl + Z]` → sends `SIGTSTP`

### View jobs

```
jobs
```

### Resume in background

```
bg
```

### Bring to foreground

```
fg <job_id>
```

### Run in background directly

```
ping -c 10 example.com &
```

---

## Execute Multiple Commands

Operator	Description
<code>;</code>	Run all commands regardless of success
<code>&amp;&amp;</code>	Only run next if previous succeeds
<code> </code>	Pipe output of one to another

### Examples

```
echo '1'; echo '2'; echo '3'  
echo '1' && echo '2' && echo '3'  
cat file.txt | grep "error"
```

---

## Summary

- Use `systemctl` for managing services.
- Use `ps`, `jobs`, `fg`, `bg` for process control.
- Use `kill`, `pkill`, `killall` for process termination.
- Use `&&`, `;`, `|` to chain commands efficiently.

---

Happy Hacking! 