# CMPU 4050: Systems Integration
# Week 1 — Introduction & Linux Foundation

Dr Stephen O'Sullivan

# Today's Agenda

1. Module overview and assessments
2. What is systems integration?
3. Linux administration essentials
    - Users, groups, and permissions
    - File system hierarchy (FHS)
    - Package management
    - System services (systemd)
    - Basic troubleshooting
4. Shell basics and automation
5. Preparation for next week

### Learning Outcomes

This lecture addresses:

**MLO1**: Describe common services

**MLO2**: Configure services

**MLO4**: Analyse system issues

# Section Overview

1. Module Overview
2. Systems Integration Fundamentals
3. Linux Administration Essentials
4. Shell Basics and Automation
5. Practical Exercises
6. Summary and Preparation

# What is Systems Integration?

## Definition

The process of bringing together diverse computer systems and software applications physically or functionally to act as a coordinated whole.

- **Core focus**: Making different systems work together seamlessly
- **Key services**: DNS, DHCP, NFS, Authentication, File sharing
- **Platforms**: Primarily Linux, with Windows integration
- **Skills**: Configuration, troubleshooting, documentation

## Industry Relevance

These skills are fundamental for DevOps, cloud engineering, and infrastructure roles.

# Assessment Structure

## Written Examination (60%)

- Covers all lecture material
- Assumes labs completed

## Continuous Assessment (40%)

- **Assignment 1** (10%): DNS Resolver
- **Assignment 2** (30%): Server Integration

## Key Dates

- Assignment 1 Submission: Week 6
- Assignment 2 Submission: Week 10
- Assignment 2 Demos: Weeks 11 (12)

## Late penalties

- Less than 1 day: 7%
- Less than 2 days: 14%
- ⋯
- Less than 1 week: 49%
- 1 week: Submissions **not** accepted

# Assessment Requirements 1

## Generative AI Policy

- GenAI permitted for learning, debugging, and concept clarification
- All submitted code and documentation must be your original work
- You must understand and be able to explain every line of your submission
- Required attribution: Document any AI assistance in a file named `AI-USAGE.txt`
- Prohibited: Direct code generation, documentation writing, or solution copying
- Validation: Oral examination may be required to verify understanding

# Assessment Requirements 2

## Submission Protocol

Both assignments must conform to these requirements:

1. Single submission via Brightspace (resubmissions overwrite previous)
2. Single zip file
3. Assignment 1 zip file: source code + doc + `AI-USAGE.txt`
4. Assignment 2 zip file: config files + doc + `AI-USAGE.txt`
5. **No bloat.** This means you must not submit backups or old versions of your files, provided helper code, binary files, `.git` folders, `node_modules`, VM images, system files, etc.

# Module Schedule

**First Half**

1. Introduction & Linux (today)
2. Networking Fundamentals
3. DNS Part 1 (Resolution)
4. DNS Part 2 (Servers)
5. DHCP
6. NFS

**Second Half**

7. (No lecture — Reading Week)
8. FTP & SSH
9. Authentication (Kerberos/LDAP)
10. Samba & Windows Integration
11. Active Directory
12. Performance & Integration

# Section Overview

1 Module Overview

2 Systems Integration Fundamentals

3 Linux Administration Essentials

4 Shell Basics and Automation

5 Practical Exercises

6 Summary and Preparation

# Why Systems Integration Matters

**Modern IT Challenges**

- Heterogeneous environments
- Legacy system compatibility
- Cloud and on-premise mix
- Security requirements
- Scalability needs

**Integration Solutions**

- Centralized authentication
- Unified file systems
- Name resolution services
- Automated configuration
- Monitoring and logging

### Real-World Example

A typical enterprise might run Windows desktops authenticating against Active Directory, Linux web servers using LDAP, network storage via NFS/SMB, with DNS and DHCP tying everything together.

# The Integration Mindset

## Core Principles

1. **Incremental changes**: Small, reversible modifications
2. **Documentation**: Every change must be documented
3. **Testing**: Verify in isolation before integration
4. **Monitoring**: Know when things break
5. **Automation**: Reduce manual configuration errors

## Before Making Any Change

- Check current state: `systemctl status`
- Review logs: `journalctl`
- Backup configuration: `cp config config.bak`
- Document your plan
- Test rollback procedure

# Section Overview

1. Module Overview
2. Systems Integration Fundamentals
3. Linux Administration Essentials
4. Shell Basics and Automation
5. Practical Exercises
6. Summary and Preparation

# Why Linux for Systems Integration?

- **Industry standard**: Dominates server infrastructure
- **Open protocols**: Excellent support for standards (DNS, DHCP, LDAP)
- **Flexibility**: Highly configurable for any role
- **Automation friendly**: Strong CLI and scripting capabilities
- **Cost effective**: No licensing for core services

## Distribution Focus

This module primarily uses Ubuntu Server (Debian-based), but concepts apply to:

- Enterprise: RHEL, SLES, Ubuntu LTS
- Community: Debian, Fedora, openSUSE
- Specialized: Alpine (containers), pfSense (networking)

# Users and Groups 1

## Core Concepts

- Users defined in `/etc/passwd`
- Groups defined in `/etc/group`
- Passwords (hashed) in `/etc/shadow`
- Primary group vs supplementary groups

```
sudo adduser alice     # Create user with home directory
sudo usermod -aG sudo alice     # Add user to supplementary group

# View user information
id alice
groups alice

su - alice     # Switch to user (for testing)
```

# File Permissions

## Permission Structure

```
-rwxr-xr-- 1 alice dev 2048
|\_/\_/\_/
| |  |  +-- other (r--)
| |  +----- group (r-x)
| +-------- owner (rwx)
+---------- type (-)
```

## Numeric Representation

- Read (r) $= 4$
- Write (w) $= 2$
- Execute (x) $= 1$
- Combined: rwx $= 7$

```
sudo chown alice:dev file.txt      # Change ownership

chmod u=rwx,g=rx,o=r file.txt      # Set permissions (symbolic)

chmod 754 file.txt      # Set permissions (numeric)

ls -la file.txt      # View permissions
```

# Special Permissions

## Three Special Bits

- **SUID (4)**: Execute with owner's privileges
- **SGID (2)**: Execute with group's privileges / inherit group
- **Sticky (1)**: Only owner can delete in shared directories

```
chmod g+s /srv/shared      # Set SGID on directory (new files inherit group)

chmod +t /tmp      # Set sticky bit on /tmp (users can't delete others' files)

# SUID example (passwd command)
ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 68208 ... /usr/bin/passwd

# Combined special + regular permissions
chmod 2775 /srv/project  # SGID + rwxrwxr-x
```

# File System Hierarchy Standard (FHS)

## System Directories

| | |
|---|---|
| /bin | Essential binaries |
| /sbin | System binaries |
| /lib | Shared libraries |
| /etc | Configuration files |
| /var | Variable data |
| /tmp | Temporary files |

## User/Application Directories

| | |
|---|---|
| /usr | User programs |
| /usr/local | Local additions |
| /opt | Optional software |
| /home | User home directories |
| /srv | Service data |
| /mnt, /media | Mount points |

### Integration Relevance

Services store configs in `/etc`, data in `/var` or `/srv`, logs in `/var/log`

# Package Management

## Debian/Ubuntu (APT)

```
# Update package index
sudo apt update

# Upgrade packages
sudo apt upgrade

# Install package
sudo apt install nginx

# Search packages
apt search dns

# Show package info
apt show bind9
```

## RedHat/Fedora (DNF/YUM)

```
# Update package index
sudo dnf check-update

# Upgrade packages
sudo dnf upgrade

# Install package
sudo dnf install nginx

# Search packages
dnf search dns

# Show package info
dnf info bind
```

# SystemD Service Management

## SystemD Components

- **Units**: Services, sockets, timers, mounts
- **Targets**: Groups of units (like runlevels)
- **Journal**: Centralized logging

```
sudo systemctl start nginx      # Service control
sudo systemctl stop nginx
sudo systemctl restart nginx
sudo systemctl reload nginx

sudo systemctl enable nginx     # Enable/disable at boot
sudo systemctl disable nginx

systemctl status nginx      # Check status
systemctl is-active nginx
systemctl is-enabled nginx
```

# Reading Service Status

```
$ systemctl status nginx
* nginx.service - A high performance web server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-09-16 10:23:45 IST; 2h 15min ago
     Docs: man:nginx(8)
  Process: 1234 ExecStartPre=/usr/sbin/nginx -t -q (code=exited, status=0/SUCCESS)
  Process: 1235 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
 Main PID: 1236 (nginx)
    Tasks: 2 (limit: 4915)
   Memory: 3.2M
   CGroup: /system.slice/nginx.service
           |-1236 nginx: master process /usr/sbin/nginx
           `-1237 nginx: worker process
```

## Key Information

- **Loaded**: Unit file location and boot status
- **Active**: Current state and uptime
- **Main PID**: Primary process ID
- **CGroup**: Process tree and resource usage

# Viewing Logs with Journalctl

```
sudo journalctl      # View all logs
sudo journalctl -f      # Follow logs in real-time
sudo journalctl -u nginx      # Service-specific logs

# Time-based filtering
sudo journalctl --since "1 hour ago"
sudo journalctl --since "2024-09-16" --until "2024-09-17"

# Priority filtering
sudo journalctl -p err     # Errors and above
sudo journalctl -p warning  # Warnings and above

# Boot logs
sudo journalctl -b          # Current boot
sudo journalctl -b -1       # Previous boot

journalctl --disk-usage      # Disk usage
```

# Section Overview

1. Module Overview
2. Systems Integration Fundamentals
3. Linux Administration Essentials
4. **Shell Basics and Automation**
5. Practical Exercises
6. Summary and Preparation

# Essential Commands for Integration Work

## Network Diagnostics

```
# Show interfaces
ip addr show
ip link show
ip route show      # Show routing

ping -c 4 8.8.8.8      # Test connectivity

traceroute google.com

nslookup tudublin.ie      # DNS lookup
dig tudublin.ie

# Port testing
nc -zv localhost 80
ss -tulpn
```

## File Operations

```
find /etc -name "*.conf"      # Find files
grep -r "listen" /etc/nginx/      # Search
    in files

tail -f /var/log/syslog      # Monitor files
diff config.old config.new      # Compare
    files

# Archive
tar czf backup.tar.gz /etc/
tar xzf backup.tar.gz

# Disk usage
df -h
du -sh /var/*
```

# Shell Scripting Fundamentals

```
#!/bin/bash
# backup-config.sh - Backup service configurations

BACKUP_DIR="/backup/configs"      # Variables
DATE=$(date +%Y%m%d-%H%M%S)
SERVICES="nginx bind9 dhcpd"
mkdir -p "$BACKUP_DIR"      # Create backup directory

for service in $SERVICES; do      # Loop through services
    if systemctl is-active --quiet "$service"; then
        echo "Backing up $service configuration..."
        tar czf "$BACKUP_DIR/${service}-${DATE}.tar.gz" \
            "/etc/$service/" 2>/dev/null
    else
        echo "Warning: $service is not running"
    fi
done
echo "Backup completed to $BACKUP_DIR"
```

# Automation with Cron

## Cron Schedule Format

`minute hour day-of-month month day-of-week command`

```
crontab -e      # Edit user crontab

# Example entries: Backup configs daily at 3 AM
0 3 * * * /usr/local/bin/backup-config.sh

# Check disk space every hour
0 * * * * df -h | mail -s "Disk Usage" admin@example.com

# Restart service weekly on Sunday at 2 AM
0 2 * * 0 systemctl restart nginx

crontab -l      # View current crontab

sudo nano /etc/cron.d/backup-job      # System-wide cron in /etc/cron.d/
```

# Section Overview

1. Module Overview
2. Systems Integration Fundamentals
3. Linux Administration Essentials
4. Shell Basics and Automation
5. Practical Exercises
6. Summary and Preparation

# Lab Exercise: System Profile

**Objective**: Create a system profile documenting your VM's configuration

```
sudo adduser svcuser      # 1. Create a new user for services
sudo usermod -aG www-data svcuser

sudo mkdir -p /srv/project      # 2. Set up a project directory with proper permissions
sudo chown svcuser:www-data /srv/project
sudo chmod 2775 /srv/project  # SGID for group inheritance

sudo apt update      # 3. Install and verify a web server
sudo apt install nginx -y
sudo systemctl enable --now nginx

cat <<EOF > ~/system-profile.txt      # 4. Document the setup
System: $(hostname)
Date: $(date)
User: $(id svcuser)
Service: $(systemctl is-active nginx)
EOF
```

# Troubleshooting Scenario

## Scenario

A service fails to start after configuration changes. Walk through the troubleshooting process.

```
systemctl status myservice      # 1. Check service status

journalctl -u myservice -n 50      # 2. View recent logs

myservice --config-test  # Service specific      # 3. Test configuration

ls -la /etc/myservice/      # 4. Check file permissions

systemctl list-dependencies myservice      # 5. Verify dependencies

# 6. Try manual start for detailed errors
sudo -u serviceuser /usr/bin/myservice --debug
```

# Section Overview

1 Module Overview

2 Systems Integration Fundamentals

3 Linux Administration Essentials

4 Shell Basics and Automation

5 Practical Exercises

6 Summary and Preparation

# Key Takeaways

## What You Should Understand

- How to manage users, groups, and permissions in Linux
- The File System Hierarchy and where services store files
- How to control services with systemd
- How to read logs and diagnose problems
- Basic shell scripting for automation

## Before Next Week

1. Complete the lab exercise (system profile)
2. Ensure your VM is working with bridged networking
3. Review basic networking concepts (TCP/IP, routing)
4. For supplementary reading, see Nemeth (Chapter 1)

# Next Week: Networking Fundamentals

## Topics We'll Cover

- TCP/IP protocol stack review
- IP addressing and subnetting
- Routing tables and packet flow
- Network address translation (NAT)
- Basic firewall concepts
- Network troubleshooting tools

## Why This Matters

All system integration relies on network communication. Understanding how packets flow between systems is essential for configuring and troubleshooting integrated services.

# Section Overview

7 Quick Reference

# Command Quick Reference

## User Management

```
adduser username
usermod -aG group user
deluser username
passwd username
id username
groups username
```

## Service Management

```
systemctl start service
systemctl stop service
systemctl restart service
systemctl enable service
systemctl status service
journalctl -u service
```

## File Permissions

```
chown user:group file
chmod 755 file
chmod u+x file
umask 022
setfacl -m u:user:rwx file
```

## Network Basics

```
ip addr show
ip route show
ping -c 4 host
nslookup domain
ss -tulpn
nc -zv host port
```

# Configuration File Locations

**System Configuration**

| | |
|---|---|
| /etc/passwd | User accounts |
| /etc/group | Groups |
| /etc/shadow | Passwords |
| /etc/sudoers | Sudo config |
| /etc/fstab | Filesystems |
| /etc/hosts | Static hosts |

**Service Configuration**

| | |
|---|---|
| /etc/ssh/ | SSH server |
| /etc/nginx/ | Nginx web server |
| /etc/bind/ | DNS (BIND) |
| /etc/dhcp/ | DHCP server |
| /etc/samba/ | Samba/SMB |
| /var/log/ | All log files |

**Remember**: Always backup before modifying!

```
cp /etc/service/config{,.bak}
```