Advanced Operating System Architecture

Lab 1: Linux Kernel Configuration and Compilation

Computer Science Department

Academic Year 2023-2024

General Information

- **Release Date**: November 06, 2024
- **Submission deadline**: November 20, 2024
- **Objective**: Learn how to download, configure, compile and install a custom Linux kernel

Prerequisites

- Linux distribution installed (Ubuntu/Debian recommended)
- At least 20 GB of free disk space
- Minimum 4 GB RAM recommended
- Root/sudo access
- Development tools (to be identified and installed by student)

1 Source Code Download

1.1 Download Methods

Student must choose and implement one of the following methods:

1. **Official Kernel Archive**
   - Locate and download the latest stable version from kernel.org
   - Extract the archive
   - Navigate to the appropriate directory
2. **Git Repository** (Recommended)
   - Locate the official Linux kernel git repository
   - Clone the repository
   - Checkout the latest stable version

Tasks

1. Document your chosen method
2. Justify your choice of method
3. Note the kernel version you're using

1.2 Source Code Structure

Explore and document the purpose of key directories:

- `arch/`
- `drivers/`
- `include/`
- `kernel/`
- `Documentation/`
- `Makefile`

Tasks

1. Create a brief description of each directory
2. Identify the location of drivers for your hardware

2 Kernel Configuration

2.1 Understanding Configuration Options

Student must research and document:

- The meaning of CONFIG_* options
- Different option types (y/m/n)
- Configuration locations

2.2 Configuration Methods

Research and implement one of these methods:

1. Using current system configuration
2. Minimal configuration
3. Menu-based configuration

2.3 Essential Configuration Options

Identify and document options needed for:

- Processor architecture
- Basic file systems
- Essential device drivers
- Core kernel features

Tasks

1. Choose and justify your configuration method
2. Document at least 10 essential options
3. Explain the necessity of each option

3 Kernel Compilation

3.1 Compilation Steps

Research and implement:

1. Cleaning previous builds
2. Compiling the kernel
3. Compiling and installing modules
4. Optimizing compilation speed

Tasks

1. Document each compilation step
2. Monitor and record resource usage
3. Note encountered errors and their solutions

4 Kernel Installation

4.1 Installation Steps

Research and implement:

1. Installing the compiled kernel
2. Updating the bootloader
3. Creating initial RAM disk
4. Configuring boot options

Tasks

1. Document the installation process
2. Verify successful installation
3. Create a backup plan for boot failure

5 Submission instructions

The following GitHub repository setup is used for submitting lab work.

Tasks

1. **Create a private repository** named **"Advanced OS"**.
2. **Add a ‘README.md‘ file** with the names of all group members.
3. **Organize work** in subdirectories (e.g., ‘Lab0‘, ‘Lab1‘, etc.) for each lab.
4. **Add the professor (balimou)** as a collaborator to the repository.

6 Verification and Testing

6.1 Required Verifications

1. Kernel modules installation check
2. Boot configuration verification
3. Successful boot test
4. Kernel version verification

7 Deliverables

7.1 Documentation (40%)

- Complete list of commands used with explanations
- Configuration choices and rationale
- Problems encountered and solutions

7.2 Evidence (30%)

- Screenshots of key steps
- Final configuration file
- Proof of successful boot with new kernel

7.3 Technical Understanding (30%)

- Written explanation of configuration choices
- Analysis of compilation process
- Implemented safety measures

Warning!

- Do not proceed to next step without instructor verification
- Keep detailed notes of all commands
- Maintain backups at each major step
- Document all error messages and their resolution

8 Evaluation Criteria

- Command understanding and proper usage (30%)
- Configuration choices and justification (25%)
- Successful compilation and boot (25%)
- Documentation quality (20%)