ChatGPT

BFS-CFS Hybrid Scheduler Simulation

Project Overview

This document outlines the simulated execution and testing process of a BFS-CFS hybrid scheduler modification in a virtualized Linux environment. It aims to showcase each major stage of the development and deployment pipeline of a custom kernel build for hybrid task scheduling.

Environment Setup

Host Machine - OS: Windows 11 (HP Victus, 11th Gen Intel Core i5/i7) - Virtualization: VirtualBox

Guest Machine - OS: Ubuntu 22.04 LTS Desktop - RAM: 4 GB - CPU Cores: 2

Estimated Time: 20 minutes

Actions: - Create VM named `Ubuntu_kernel_test` - Mount Ubuntu 22.04 ISO - Install OS and perform updates:

```
sudo apt update && sudo apt upgrade -y
sudo apt install build-essential libncurses-dev bison flex libssl-dev libelf-dev
```

Kernel Download and Preparation

Estimated Time: 25 minutes

```
wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.15.1.tar.xz
tar -xvf linux-5.15.1.tar.xz
cd linux-5.15.1
make defconfig
```

This prepares the kernel source for modification.

Kernel Modification: Implement BFS-CFS Switch

Estimated Time: 30 minutes

Files Involved: - `kernel/sched/core.c` → modified for dispatching logic - `kernel/sched/fair.c` → holds default CFS code with added switch logic - `kernel/sched/bfs.c` → new file introduced with BFS logic

Key Features: - Tasks marked via `nice` value or flag can enter BFS mode - BFS allowed only on CPU 0 and 1 - 15-minute timeout auto-switches BFS tasks back to CFS

```
git init
git add .
git commit -m "Add BFS-CFS hybrid scheduling logic"
```

Kernel Compilation and Installation

Estimated Time: 40 minutes

```
make -j$(nproc)
sudo make modules_install
sudo make install
```

A new entry is created in the GRUB bootloader.

Boot and Test Custom Kernel

Estimated Time: 10 minutes

```
sudo reboot
```

Use the GRUB menu to select the new kernel.

Runtime Verification and Task Testing

Estimated Time: 30 minutes

- Run sample load scripts.

- Assign tasks manually to BFS via `nice` levels.
- Confirm correct CPU-core mapping.
- Observe fallback behavior to CFS.

Commands Used:

```
dmesg | grep "BFS"
ps -eo pid,psr,comm,ni,pri | grep test_task
```

Total Simulation Time: ~2.5 to 3 Hours

This simulation assumes continuous execution without breaks. Timings may vary based on system performance and configuration.

Notes

This PDF is intended as part of a GitHub repository for academic or demonstration purposes and can be used to explain hybrid scheduling models in OS development or kernel engineering contexts.