MSCO3

Saint Leo University

Advanced Operating System

JVCRR Based Scheduler in Linux

NARENDAR REDDY KARNATI

Student ID: 001177351

E-mail: [narendarreddy.karn@email.saintleo.edu](mailto:narendarreddy.karn@email.saintleo.edu)

AND

RAJU SAMALA

Student ID: 001173818

E-mail: [raju.samala@email.saintleo.edu](mailto:raju.samala@email.saintleo.edu)

Abstract:

Open-source software and free operating systems have piqued the curiosity of many individuals. People have failed to notice that while downloading standard distributions, users are usually pushed to install a significant number of software that they will almost certainly never use. These projects are a complete waste of money.

In addition, high-complexity scheduling algorithms such as Weighted Fair Queuing are employed in Linux.

It is hard to prevent the insertion of components that will never be used unless the Linux OS can be altered. Small Linux systems may be built using the Linux From Scratch (LFS) technology. Setting up a Linux system, developing and installing individual packages, customizing boot scripts, and installing the kernel are all covered by LFS. Process scheduling may also be utilized to respond to changing needs, such as the need for faster execution. The LFS system may be used to implement the Jumping Virtual Clock Round Robin technique, which combines Dynamic JVC and Static RR to overcome the limitations of conventional Weighted Fair Queueing and Leaf Forward Virtual Clock algorithms.

Introduction:

A Linux-based distribution is made up of the Linux kernel and a collection of applications. The Linux kernel contains a scheduler module since it is modular. The scheduler module is built using the scheduling algorithm. The scheduling approach employed in Linux kernel version 2.6 is known as fair scheduling. Fair scheduling does not guarantee that all work will be done in a timely manner[1]. As a consequence, the proposed research employs a unique hybrid scheduling approach known as Jumping Virtual Clock Round Robin (JVCRR), which combines three scheduling algorithms: virtual clock, jumping queue, and Round Robin (RR)

Because the scheduler developed cannot be directly incorporated into the Linux kernel due to the possibility of several malfunctions that are difficult to identify and understand at the development level, a Linuxsche tool will be used to verify that the scheduler's functionality is correct in all aspects. The Linuxsche tool is an open source simulator that will give the same environment for checking the simulator's correctness.

The second part of the project is about the packages that are used to build the operating system on top of the Linux kernel, which is subsequently included in Linux distributions using an open-source tool called LFS. LFS is a collection of procedures that must be followed in order to build a Linux distribution.

LFS offers assistance with every area of Linux distribution development. A new Linux scheduler for Linux kernel version 2.6, as well as a minimal Linux-based distribution for basic computing needs, are being created as a consequence of this project.

LFS is a project that offers every developer instructions in order for them to construct their own customized distribution. Because of its modular nature, Linux is the most versatile operating system available, and it can be used on everything from a modest personal computer to a supercomputer. LFS is mainly concerned with the compilation of a large number of Linux packages that are available in various repositories throughout the world and brought together over the internet.

The efficiency of any operating system is mostly influenced by how rapidly it completes tasks. The scheduler module in an operating system distributes tasks to the kernel based on priority or the shortest job. An operating system's scheduling algorithm selects which task will execute next. Because there are so many scheduling algorithms to choose from, finding the appropriate one is crucial.

The scheduling techniques available include Weighted Fair Queuing, Virtual Clock, Round Robin, and Jumping Virtual Clock, all of which are inefficient. Because each of them requires a certain amount of time to carry out each order. All of the previous approaches are useless due to their complexity of roughly O(log N). Any OS that promises to be more efficient and better should have a complexity of at least O (1) [3].

Techniques like Virtual Clock and Leaf Forward Virtual Clock (LFVC) have O(1) complexity for calculating the process' virtual time. However, choosing the next queue requires complexity O(log N) and O(log N), which is inadequate for any faster operating system[3].

The present Linux kernel uses fair scheduling, which is efficient but does not guarantee that programs will be performed at O. (1). As a consequence, a novel scheduling approach called as JVCRR is used to improve the overall efficiency of the Linux system by ensuring an O(1) rate.

JVCRR is a hybrid approach that combines the Round Robin and Jumping Virtual Clock scheduling methods. This technique includes both calculating virtual time and selecting the next queue. The total complexity is O(1) since both calculating virtual time and choosing the next queue require O(1) time (1). As a consequence, JVCRR beats the other strategies for scheduling.

Plan:

Diagram

Description automatically generated

Text, letter

Description automatically generated

A picture containing company name

Description automatically generated



Text, letter

Description automatically generatedChart, line chart

Description automatically generated

References:

Jonathan Corbet, Amanda McPherson, ‘Linux Kernel Development’, The Linux Foundation, http://www.linuxfoundation.org, Dec 2010. Masrurkhah, Alireza Abed, Amir Seyed Danesh, and Seyyedeh Narjes Ghiami Taklimi, ‘A survey on implementation of a Linux-based operating system using LFS method’ International Journal of Computer Science Issues (IJCSI) 9, no. 2 (2012). Hwang, Lain-Chyr, Steen J. Hsu, San-Yuan Wang, and Yong-Hua Huang. ‘A hybrid scheduling algorithm with low complexity: Jumping Virtual Clock Round Robin’, In Distributed Computing Systems Workshops, 25th IEEE International Conference on, pp. 698-703. IEEE, 2005. Sukumar B. B, Neelima P. N, and Sunil. Kumar.B, ‘Efficient Round Robin CPU Scheduling Algorithm’, volume 4, Issue 9, pp. 36-42, November 2012. Robert Love ‘Linux Kernel Development’, Third Edition, ISBN-13: 978-0-672-32946-3, June 2010. Gerard Beekmans, ‘Linux From Scratch’, Version 7.4, 2013. Silberschatz, 2.P. Galvin, and 3.G. O. Gagne, "Operating System Concepts", 7th Edition, John Wiley and Sons. INC, 2005