ADVANCED OPERATING SYSTEMS

TERM PROJECT WEEKLY REPORT

Under,

An-I Andy Wang

Shuanglong Zhang

By,

Akshara Dendi ([ad13r@my.fsu.edu](mailto:ad13r@my.fsu.edu))

Lehari Saggam ([ls13d@my.fsu.edu](mailto:ls13d@my.fsu.edu))

Sinduri Shyamala ([ss13w@my.fsu.edu](mailto:ss13w@my.fsu.edu))

**OPTION CHOSEN:** SURVEY PROJECT

**PROJECT TITLE:** A STUDY ON LINUX LOCK SYSTEMS.

**GOAL:** To do a detailed analysis on the Linux lock systems.

**TEAM MEMBERS:** Akshara Dendi ([ad13r@my.fsu.edu](mailto:ad13r@my.fsu.edu))

Lehari Saggam ([ls13d@my.fsu.edu](mailto:ls13d@my.fsu.edu))

Sinduri Shyamala ([ss13w@my.fsu.edu](mailto:ss13w@my.fsu.edu))

**AKSHARA DENDI’S CONTRIBUTION:**

I have gone through the history and background of kernel patches and its features in different operating system which included different editions of kernel patches and protection required. There are many reasons why new patches come into existence and one of them is to provide scalability and reliability in kernel. The kernel have different privileges in different operating systems which leads to many advantages and disadvantages over each other’s and there are also many criticism and weakness of kernel patches.

My survey for this week also included about patches and its uses. New patches are the modifications or small changes to already existing patches for some advantage over previous ones. The difficulty in patching is to get an insight of how they are implemented, applied or how the source tree and files are feed to them. There are also many error handling available and also alternates to patches.

I have also gone through Linux kernel study, its history over ages and the rapid increase of its usage. The technical features of Linux support many activities and it as a monolithic kernel. It is written in c languages and earlier version of Linux were not portable but trend changed over the years to portability. Different kernel versions have been evolved and the latest version is 3.X. Many changes have been noticed and the rate of change is continuously increasing.

References:

[1] “Linux Kernel Development” by Greg Kroah-Hartman, SuSE Labs / Novell Inc. Jonathan Corbet, LWN.net Amanda McPherson, The Linux Foundation

[2] “IEGD Linux Kernel Module - Porting and Patching Methods” by Mudit Vats- Software Engineer Intel Corporation, Ryan Lovrien- Software Engineer Intel Corporation

[3] “On submitting kernel patches” by Andi Kleen Intel Open Source Technology Center

[4] <http://en.wikipedia.org/wiki/Kernel_Patch_Protection>

[5] <https://www.kernel.org/doc/Documentation/applying-patches.txt>

[6] <http://en.wikipedia.org/wiki/Linux_kernel>

**LEHARI SAGGAM’S CONTRIBUTION:**

I have started reading the papers related to Linux as I wanted to start from the basics to become familiar with Linux operating systems, its history and some basic amenities like its scalability issues, its security aspects and support given by them for Linux kernel.

**PAPERS READ:**

* A Study of Linux File System Evolution
* The Multikernel: A new OS architecture for scalable multicore systems
* An Analysis of Linux Scalability to Many Cores
* Linux Security Modules: General Security Support for the Linux Kernel

**INFORMATION GATHERED FROM PAPERS:**

The paper “A Study of Linux File System Evolution” talks about the study of the file system code and its evolution. The paper responses to some crucial issues like reliability enhancements, complications of file systems, supreme bug types, performance optimizations and etc.., When a new patch is added to the existing version then it shows that there is a new evolution and the current version is being changed to higher version though the before version and its patch is available. The authors have done a manual patch inspection. They have done a significant study on patch types, performance, bug patterns and their consequences. The main purpose of the patches is to fix various aspects like performance, bugs and thereby increasing the reliability and adding new features. The paper further shows the techniques used to build performance in file systems. Improving synchronization, scalability, locality, access optimization, scheduling and etc.., are the techniques that are commonly implemented in all the file systems (XFS, Ext4, Btrfs, Ext3, Reiser, JFS). The paper is very useful when it comes to gaining knowledge in Linux file system and it provides us with comparisons, case studies, related works and rich data for further research.

In the paper “The Multikernel: A new OS architecture for scalable multicore systems”, the authors have explored a new OS structure called “multikernel”. The multikernel treats the system as a network that is of independent cores taking into consideration of no inter-core sharing at the minimum level. The authors introduced the design fundamentals of accurate communication, hardware-neutral structure, and state replication. They proposed a multikernel, Barrelfish, which analyzes the application of the model to a detailed OS implementation. They showed the assessment that Barrelfish satisfies the objective of scalability and adaptability to the hardware aspects thereby giving the competing performance on a new hardware. The paper discusses multikernel “Barrelfish” which can scale good with the core count for these operations. It can easily get acquainted to implement more competent communication patterns and demonstrate the importance of the pipelining and batching of request messages without changing the code of the OS which performs the operations.

The paper “An Analysis of Linux Scalability to Many Cores” talks about the scalable applications on Linux. The paper gives us the details on operating system kernel as many applications spend time in the operating system kernel. The paper shows the analyzing of scalability by using an off-the-shelf 48-core x86machine and through this they scale a set of applications by maintaining a constraint of parallel implementation and being system intensive. The main contribution of the paper is that they have analyzed the Linux scalability for 7 real applications. They have stocked the Linux limit scalability and have analyzed the bottlenecks of the analysis. The paper discusses about the Linux spin lock implementation and its technique.

The paper “Linux Security Modules: General Security Support for the Linux Kernel” talks about the designing and implementation of Linux security modules and explains the challenges in giving an optimal solution that leaves the least impact on the Linux Kernel. The Linux Security Modules (LSM) plan has built a lightweight, general purpose, access control framework for the dominant Linux kernel that facilitates various different access control models to be implemented as loadable kernel modules. Various existing enhanced access control implementations which include Linux capabilities, Security- Enhanced Linux (SELinux), and Domain and Type Enforcement (DTE) are already making use of the LSM framework.

**OBSTACLES ENCOUNTERED**: The main obstacle encountered in our project is that we do not have much data or any open source which talks about the locks in Linux systems. There are very less papers which talk about them and we do not have a clear description on them in any of the papers.

My next task would be to read any material which talks about the series of locks in Linux systems which are Global lock, Semaphores and Mutex. I will be going through the papers which have effective information about the locks.

**NEW IDEAS:**

We have researched about some of the patching tools used in Windows and other operating systems. We are planning to use a tool for finding the patch in series of codes. This can gradually decrease our work load over finding patch in each code. Since, there are almost 40 to 50 patches in each lock system and we have to in-depth survey on each patch in semaphores. We will be determining the patch in the code and will be analyzing its code and the reason why the extension was made to the previous series.

We are planning to show our results in the format of a graph reading for the performance of the locks. Since, this will be providing a better understanding and a better insight of the project.

**SINDURI SHYAMALA’S CONTRIBUTION:**

**Paper 1- ffsck: The Fast File System Checker**

Basically huge amounts of data can be divided into chunks of data and every chunk can be called a “file”. Such files that are created are prone to damage and hence we have seen that “checkers” have been introduced. A small damage in the data causes a great loss to the entire data and hence checkers were developed and these checkers have been protecting the file systems. The latest one is the fast file system checker, which was introduced recently. We have read a research paper about the fast file system checker. This type of file system also has a few disadvantages.

**Paper 2- Fault Isolation and Quick Recovery in Isolation File Systems**

The next paper that we have read is about fault isolation. Now when there are errors in any particular file or code, we know that they should not be overlooked. In order to avoid these errors, a fault isolation system has been built and this protects the files from being entirely damaged by using the concept of “file pods”. In every file pod, there are independent domains that have their own repair policies whenever there is any damage in the system. So the entire system need not be considered. For better isolation, journaling has been introduced. We have been trying to read more about the evolution of the file systems. The older versions have been gradually replaced with the newer ones due to the speed, flexibility, differences in patches and the type of locks used. By the end of the project, we will come to a conclusion about the various versions that have been used and the reasons for the introduction of newer versions.

**Paper 3- A Study of Linux file system evolution**

We have read the paper on file system evolution and we have been trying to understand the reasons for the many different versions that were created. A difference in the patches is an important aspect of our project and we have read about the basics of patches and how they are used. Also we have read and understood about the bugs and the different types of patches that have been developed. The performance and reliability of the patches have to be improved in order to use them in a better way. We will continue our study on the patches and also try to figure out the various differences in the patches.

**Paper 4- Box: Towards Reliability and Consistency in Dropbox-like File Synchronization Services**

Another paper that we have tried to analyze is the “\*-Box” which has been important for cloud storage of data. Gradually the file systems are being replaced by the cloud storage. Various cloud storage drives such as Dropbox, Google drive have become very popular. Inspite of these drives being so popular and considered to be safe, they are not actually very safe. So measures have been introduced by researchers about how to achieve good reliability and consistency for file synchronization. These papers have helped us to understand the importance of file systems and how new measures have been introduced to protect the data without being damaged. Also we have been able to analyze the importance of using file systems for data storage using dropbox and the various steps that have been taken to make the use of dropbox better. Methods about how to recover from crashes were introduced by the authors of the paper.