Archer Linux

Project Director:

Mr. Saud Bin Tahir

Project Supervisor:

Mr. Muhammad Bilal

Submitted by:

Muhammad Asad Noman (BSCSF16MM001)
Qadeer Ahmad (BSCSF16MM026)
Muhammad Wajahat Munir (BSCSF16MM036)

Department of Computer Science and Informance Technology
University Of Mianwali, Mianwali

December, 2019

1 Introduction

A new open-source Linux distribution, Archer Linux, is being proposed, which will focus minimalism, stability, performance and user-friendliness and target desktop users. The whole system is intended to be put together in a systematic bottom-up approach to ensure minimalism. Each component in the system will be very carefully hand-picked. The system will also contain components developed from scratch to work integratedly with other components. Those components will ensure stability, unbloated system, performance and power efficiency.

2 Problem Description

When it comes to choosing among various Linux distributions, many users confess distro-hopping (changing distributions, simply because it did not fit their taste). Archer Linux proposes to bring features, of sheer importance, scattered in different distributions, under one system. The four essential features, namely minimalism, stability, performance and user-friendliness have been described in detail in upcoming sections. Other distribution happen to conflict these features. For instance, distributions like Ubuntu, Debian etc. have user-friendliness but it comes at the price of minimalism (i.e. system occupies more than 1GB of RAM with no running tasks), distributions like Arch Linux and its derivatives suffer from unstability and no cache management and needless to say that proper power management is totally neglected in the whole Linux distribution ecosystem.

3 Features

3.1 Minimalism

The system will feature lightweight graphical user interface (called openbox) with all important components carefully plugged in and configured to provide a user-friendly desktop. The system will be built minimum number of components (each chosen carefully) and redundancies will be avoided. This will ensure faster boot up time and will smoothen the execution of user's processes. The system on reaching graphical desktop is expected to occupy less than 200 MBs of RAM which in comparison to other distributions is marginal.

The graphical desktop will be very minimalistic and user-friendly with a re-configurable keybinding system for keyboard-centric users and a tablet mode users with a touch-screen available.

3.2 Stability

A native package manager called "Pac" is proposed with both graphical and command-line interfaces. This package manager will ensure three things, dependency resolution, stability and unbloatness. Following is a little description of how this package manager will cover these three features.

Dependency resolution

On removing an installed package, the system will resolve any conflicting dependencies, so that no other package relying on the same dependencies is effected. This will ensure that the removal of one package does not effect the working of other packages.

Stable system

Stability comes as an additional benefit to the system as a result of dependency resolution, which will ensure that interdependent packages don't stop working when removed. Another feature that will cover system stability is keeping track and clearing orphan packages. Orphan packages are the packages left installed in case of conflicting dependencies. User will have the ability to clear these packages manaually or automatedly by specifying a period of time.

Unbloatness

Package manager will keep track of orphan packages as well as the overall cache generated. User will be able to make a choice whether clear them manually or by specifying timers.

3.3 Performance and power efficiency

It has been covered in many online articles that the entire linux distribution ecosystem is totally silent about power saving features even though the underlying linux kernel does provide the capability. A component for Archer Linux is being proposed, Archer Advance Power Management (or simply AAPM), which will provide the ability to utilize those features. Six advance hardware level features have been picked and can be tweaked using AAPM. For users who are not well-versed of system options, all these features will be put on a scale from 1 to 5 where 1 represents most performance and 5 represents most powersave, so that the user simply have to move the scale for required performance. Each of these six features is given a description below.

CPU frequency scaling governer

This feature allows the CPU to scale its frequency up and down based on the given tweaks. There are two drivers available for CPU frequency scaling, one is called 'cpu-freq' (Wysocki, 2017) which is for older hardware and 'intel-pstate' (**intelpsate**) which is for newer hardware. Each of these have tweakable set of options which in turn scale the CPU's frequency.

Hardware controlled frequency scaling

This is essentially an extension of the above feature. This given the scaling property control to the CPU itself which quickens the process of frequency scaling. This feature is available in many modern PCs and therefore part of the configuration of AAPM.

Turbo Boost

Again a feature in modern hardware (Intel Corporation, n.d.), allows the CPUs to boost their performance for peak loads. Turbo boost enleashes much performance but also is not power-efficient (Charles, Jassi, Ananth, Sadat & Fedorova, 2009), and therefore is part of AAPM configuration.

Scheduled powersave

This feature enables the minimization of number of cores/hyper-threads used under light load condition, a feature very important to users whose computers are not performance-centric.

Energy vs performance policy

This routine defines a standard policy for the all CPU power and performance feautures. Altering this effects the value of features mentioned above, therefore it is also part of AAPM configuration.

PCIe ASPM

It is Active State Power Management for PCI express bus (Active state power management (ASPM) to reduce power consumption by PCI express components, 2012). The relative serial link is powered down if there is not traffic across it, and therefore saves power.

Runtime PM

Runtime power management is for PCIe bus devices (I/O and others) and performs the same functionality as PCIe ASPM for those devices (Stern & Wysocki, 2014).

4 Tools and technology

Every code contribution to the project will be written in either C or Python3. We will make sure while putting things together that no component in the system has dependencies on Python2 since it is to be deprecated in Jan, 2020 (Rossum, 2018). For compiling Linux Kernel and systemd etc. we are required and will use GNU make, meson, bison, flex, C compiler (gcc) and other tools for building and compiling those components. (Mauch, Boldt & Sigala, n.d.) provide an exhuastive list of these tools.

5 Methodology

The system will be built using a bottom-up approach, which means, each basic component will be compiled and then integrated with other basic component testing how the integration goes. This procedure will help build the system minimalistically. Once a working system is achieved, we will scribe down all the steps, write a script (probably in python) which will work out those steps everytime it is executed. This will automate the process of creating next version of the system (ISO file to be released).

The components like package manager and AAPM etc. will be developed totally in a modular and object-oriented fashion. Any CPU-centric or network-centric operations will be put in separate threads to smoothen their execution. Any required data will be stored as JSON objects and will be parsed when needed into their respective objects defined in Python or C code, which then can be manipulated as needed. These components will only be using standard Python and C libraries and no third-party libraries so to not burden the system.

References

Charles, J., Jassi, P., Ananth, N. S., Sadat, A. & Fedorova, A. (2009). Evaluation of the intel® core™ i7 turbo boost feature. In *2009 ieee international symposium on workload characterization (iiswc)* (pp. 188–197). doi:10.1109/IISWC.2009.5306782

Intel Corporation. (n.d.). Retrieved from https://www.intel.com/content/www/us/en/architecture-and-technology/turbo-boost/turbo-boost-technology.html

Mauch, J., Boldt, A. & Sigala, A. (n.d.). Minimal requirements to compile the kernel. Retrieved from https://www.kernel.org/doc/html/v4.15/process/changes.html

Rossum, G. v. (2018). Update python 2.7 eol date. Retrieved from https://github.com/python/devguide/pull/344

Stern, A. & Wysocki, R. J. (2014). Runtime power management framework for i/o devices. Retrieved from https://www.kernel.org/doc/Documentation/power/runtime_pm.txt

Teoh, P. T. (2012). *Active state power management (aspm) to reduce power consumption by pci express components*. US9632557B2.

Wysocki, R. (2017). Cpu performance scaling. Retrieved from https://www.kernel.org/doc/html/v4.12/admin-guide/pm/cpufreq.html