Project Proposal

Matthew Pietracupa, Jonathan Markovic & Iulia Apintilioaie

Vanier College

420-321-VA: Unix

Tassia Camoes Araujo

Thursday, April 17

1. Project description/goals:

We aim to achieve an easier alternative to using the Arch distro. Our main goal is to plan to make installing packages and overall desktop management easier for the user. The end goal would be to have an easy-to-navigate desktop environment with a variety of other apps. This new user-friendly setup would benefit the average beginner computer science student who might not be familiar with Linux.

2. Platform of choice:

Our platform of choice is a desktop platform using the Arch distro.

3. Demonstration plan:

We would be using a VM to demonstrate our project. Doing a fresh install of Arch to demonstrate how our script can make it easier to customize.

4. Requirements:

Our team plans to accomplish this project through various smaller categories. We are sure that by implementing the following, we would have succeeded in reaching our goal: a package manager, a desktop environment, a code/text editor, a browser, a file manager, a terminal emulator, a network manager, a utility tool, a font manager and a window manager. In addition, we will have to implement any needed dependencies. We will additionally add the option for users to have a systemd operation done to see if any updates are available for the system.

5. Major technical solutions compared:

The goal of the project is to make a package that is installable on any desktop arch platform that can configure their system to provide the best of the arch platform that you wouldn't normally get with just the regular setup script. This script will have easy-to-use select prompts to allow the user to fully customize their system.

Firstly, we had to decide between pacman and yay for our package manager. Pacman is Arch's official package manager, which may be harder to get up and running initially. It only installs packages from the official repositories, which are officially maintained by Arch Trusted Users. While yay can also install packages from these repositories, it is also capable of pulling them from the Arch User Repository. These packages can be maintained by basically everyone, and thus may not always be up to date, of high quality, or even free of malware.

Secondly, we had to decide between using Bash or Python for our scripts. Bash is ideal for scripting and automating everyday system tasks, especially on Linux, which is something fundamental for this project. Bash is perfect for quick scripts and chaining commands. In addition, it runs the commands directly in the shell, making it faster for file operations and system calls. Though Python is more structured to some extent, it can be more difficult for simple tasks, as it needs extra modules or subprocesses. Finally, Bash comes pre-installed on most Unix-based systems, while Python might need version management or virtual environments.

6. Timeline:

In the first week, our team will agree on the system setup choices we will utilize. We plan to finish the main logic needed to set up the user experience and add any pacman/yay commands to download that might be needed from the user's prompt. By the second week, our team will be working on any compatibility issues between the software and implementing error handling to streamline the user experience. By this time, we'd also have chosen a package manager. By the last week, we plan to fix any last-second bugs and, if time allows, add optional aesthetic elements. Since we plan to present our project during this final week, our team will have to prepare our speeches.

First Week

☑ ~~Did we agree on the system setup choices we will utilize?~~	Yes, we agreed that we are using a fresh Arch install in a VM, developing a script to install and configure essential tools such as the desktop environment, package manager, editor, browser, and more. We will be using GitHub for version control. Our goal is to make Arch more beginner-friendly for new computer science students.
☑ ~~Did we finish the main logic needed to set up the user experience?~~	Not fully, we started setting up the main user experience and got the intro screen and menu working. The basic layout is present, but I'm not fully completed yet. We still need to finish linking the menu choices to their actions so everything flows properly.
☑ ~~Did we add any pacman/yay commands to download that might be needed from the user's prompt~~	Yes, we decided on pacman since it is the official package manager for Arch Linux. It is already installed by default on every Arch system. We don't need AUR early on since all the basic packages are in the official repos such as: base, grub, vim and networkmanager. We started by trying out

	yay but it is only useful after the system is ready while pacman can do 100% of the tasks at install time. In addition, pacman uses signed packages from trusted sources while yay depends on user-contributed PKGBUILDs, which can be a big security risk if used too early since the packages can be unstable.

Second Week

☑ ~~Did we finish the main logic needed to set up the user experience?~~	Yes, we developed many scripts to handle various system components, such as installing key packages, configuring the desktop environment, and providing users with interactive prompts for customization, all while informing the user of their choices. The core structure is functional, and the scripts are easy to follow. Most of the main logic is complete and integrated into the setup process.
☑ ~~Did we work on any compatibility issues between the software and implement error handling to streamline the user experience?~~	Not entirely, Iulia implemented basic error handling to check if Yay is installed. However, some team members believe it's safer and more standard during initial setup to use Pacman. Which means we will have to either replace it or add it as well. Further compatibility and error-handling mechanisms are being discussed and will likely evolve depending on our final choice of package manager.
☑ ~~What challenges have we faced so far?~~	One of the main challenges has been dealing with infinite loops when calling scripts or functions from within other scripts. If the logic is not structured properly, some menu actions re-trigger the same function, causing an infinite loop. Another issue has been ensuring consistent behaviour across different VM since network or display components can behave differently.
☑ ~~What skills or concepts have we learned that I didn't know before?~~	We learned how to utilize Bash scripts more effectively by using functions and separate files. We also learned to better understand

	and control flow in shell scripts to prevent logic errors and infinite loops. Additionally, we became more comfortable with virtual machine environments and systemd service management.

Third Week

☐ Did we fix any last-second bugs?	Explain
☐ Did we add optional aesthetic elements?	Explain
☐ How well does the project align with our initial goals or vision?	Explain
☐ Do we think there are areas we could've improved on?	Explain

7. Team composition:

Matthew Pietracupa, Jonathan Markovic & Iulia Apintilioaie.

8. Baseline Arch Linux installation:

The baseline Arch Linux installation can be pretty complicated for users who aren't familiar with it. They are ready to start once they have downloaded the Arch Linux ISO file.

This guide does a really good job of helping with the installation process, but it still takes a while to go through, even with the guide.
https://gist.github.com/mjkstra/96ce7a5689d753e7a6bdd92cdc169bae#preliminary-steps

The first few steps are just to prepare for the rest of the installation.

- Set the keyboard layout
- Test for internet connection(easier to do if using a hardwired connection)
- Check the system clock

Now they are ready for the actual installation process

1) Disk partitioning
2) Disk formatting

3) Disk mounting
4) Package installation
5) Fstab(tells the system how to mount file systems on startup)
6) Switch to the new system (Arch-chroot /mnt)

This is the complicated part out of the way now it’s time for configurations

1) Set time/timezone
2) Set the locale (keymap)
3) Hostname & Host configuration
4) Set up root password and users
5) Grub configuration
6) Unmount everything and reboot system
7) Log in to account
8) Installing drivers and graphical environment

Each of us has tried the installation and is aware of the various difficulties one might encounter throughout their installation process. Here were 2 main issues we thought needed improvement on:

1. Arch does not have an installer GUI.

This means you have to manually use the terminal to partition, format, and mount your disk. Other installations include the base system, generating fstab, chroot, setting the timezone, locale, and hostname, and setting up a bootloader.

2. Connecting to Wi-Fi manually

Connecting to Wi-Fi during the Arch installation can be especially difficult if you’re on a laptop or don't have a wired connection. Users must rely on command-line tools. This is not easy for beginners and may fail without even giving clear error messages.

3. Lack of guided post-install configuration

After rebooting into a new Arch system, users are left with a bare minimum environment and must install everything manually, from the desktop environment to networking tools. There’s no guided setup or configuration utility to help users through things such as setting up a display manager, installing fonts, or enabling essential services. For new users, this can be a risk of breaking the system if dependencies are not managed correctly.