Nautilus: Integrate file-roller and compressed file management

- **What is your e-mail address and IRC nick?**

My e-mail is **razvan.ch95@gmail** and my IRC nicks are **razvan** (desktop) and **razvan_mobile** (phone, almost always available).

- **What is your web page, blog, or Twitter?**

My blog page is https://razvanchitu.wordpress.com/.

- **What city and country will you reside in during the summer?**

During this summer I will be staying in my hometown, **Bucharest**, in **Romania**.

- **What is your academic background, including university, major, and expected or achieved graduation date for any degree you have or are pursuing?**

I am a second year Computer Science Bachelors student at **University "POLITEHNICA" of Bucharest** (Romania, GMT + 2). Expecting to graduate in June 2018.

- **Who is a possible mentor for the project you are proposing?**

The announced mentors for this project are **Carlos Soriano** and **Cosimo Cecchi**.

- **What is the ultimate goal of your proposal?**

The goal of this project is to simplify working with archives by integrating the core features of **file-roller** in **Nautilus**, making decompression the default action for when opening a compressed archive. Alternatively, navigation through the compressed file like a normal folder would be possible as well. Compression and decompression will be handled internally by **Nautilus** using **gnome-autoar**, a library for automatically creating and extracting archives.

- **What components/modules will the proposed work modify or create?**

This project will rely heavily on **gnome-autoar**, so a first step is to make sure that it is stable, smooth and easy to use by **Nautilus** and other applications alike. The file manager will have compression and decompression operations handled internally. Following **Nautilus**, other applications (such as **Epiphany**, **Empathy**) could also have archive support added to them.

- **What benefits does your proposed work have for GNOME and its community?**

This project aims to make compression and decompression a transparent process to the user, minimizing interaction with archives. In the vast majority of cases, compressed files are just an intermediary point between the user and actual content. It is desired that the user will have full access to this content right from the start, rather than being given the indirect means available now (for example, **file-roller** selectively decompresses files and stores them in a hidden cache). Also, since dealing with archives is actually dealing with files, the user will be able to do this from the file manager.

From the developer's perspective, moving archive support to a library will allow for compression and decompression to be handled internally by applications. This should reduce the usage of **file-roller** so it can eventually be removed as an application.

- **Why are you the right person to work on this project?**

For the past months, contributing to **Nautilus** and the modules around it has been the highlight of my experience - lots of interesting things learned while coding and a great interaction with the community. Naturally, I want to take this experience to the next level, so I am very enthusiastic about the possibility of working on my own project. In particular, the idea behind this project is one that has a special significance for me, since it is what I wanted to do when I came across **Nautilus**. My passion for it and also knowing the impact it will have on the user experience are what makes me committed to give my best to working on this project.

- **How do you plan to achieve completion of your project?**

Before the working period starts

In this time I will continue contributing to **Nautilus** as usual. To have a good, detailed context of the project, I will also:

- analyze how compression and decompression are used in Nautilus, so the standards for gnome-autoar can be set
- analyze how file-roller handles different use cases
- determine needed fixes to gnome-autoar
- discuss this and other implementation aspects with the mentors

23 May - 27 May - Official GSoC start

- add modifications to gnome-autoar
- extend the test suite and cover corner cases

30 May - 10 June

The first feature to be implemented in Nautilus is decompression, making it the default action for archives. I will follow these steps:

- implement decompression as an asynchronous operation in Nautilus
- create tests to ensure the reliability of the communication between Nautilus and gnome-autoar during decompression
- make Nautilus special case archives to use internal decompression instead of application activation based on mime types

13 June - 17 June

- supply progress feedback data in case the operation takes long
- integrate decompression progress feedback in the system used by the other file operations
- implement UI for decompression, such as error messages or warning dialogs when an operation takes long

20 June - 1 July - Mid-term Evaluation

- investigate the cases when decompression should not be done implicitly, like large archives
- implement a reliable way to detect large archives. At first, this should only take into consideration the compressed file's size and possibly available disk space. Further on, we can also investigate time as a limiting factor
- investigate and implement a way to handle big archives. The expected solution is to add a dialog for opening them in file-roller
- add UI preference for disabling implicit decompression
- make sure opening in file-roller works as before when implicit decompression is disabled

4 July - 8 July

The next feature to be added to Nautilus is compression handling. This operation is very similar to the ones already handled by Nautilus (like moving, copying and deleting files) so its implementation is pretty straight forward.

- implement compression as an asynchronous operation in Nautilus
- handle compression actions internally
- integrate compression progress feedback in the system used by the other file operations

11 July - 15 July

- replace file-roller extension for selecting compression preferences with an internally managed widget
- implement UI for compression, such as error messages or warning dialogs when an operation takes long
- write tests to ensure the new compression implementation works as expected

18 July - 22 July

After adding compression and decompression support, I plan to use one week to:

- review the work done so far with the mentors
- test the new system in corner cases, looking for possible oversights
- discuss with other application maintainers about how gnome-autoar could be integrated in their projects

25 July - 5 August

At this point, Nautilus can serve as an example of how to implement compressed files management internally. In this period, I plan to:

- investigate the previous attempt of integrating gnome-autoar in applications
- improve existing solutions because they might not work in the current state of the projects
- where necessary, provide new patches for adding compression and decompression support to applications
- continue testing the new system in Nautilus and fix any issue that is noticed

8 August - 12 August

The last step is to provide Nautilus support for browsing compressed files, for which the gvfs “archive” backend would be a good solution. I need to:

- analyze the archive backend and discuss options with gvfs maintainers. Key needs are:
    - a “mount-on-demand” feature that is flexible, allows quick and transparent archive browsing and does not stall the file-manager UI
    - the possibility to display content in a user-friendly way without much effort from the Nautilus side
- implement the needed features in the archive backend, if possible, or else try to find a different solution for previewing compressed files

12 August - 23 August

- implement basic archive browsing functionality in Nautilus
- use the archive browsing feature in the case of long decompression operations
- make the pathbar allow easy navigation within and outside of an archive

Plans for the future

My experience so far with Nautilus has shown me that even what seems flawless in the beginning can end up having issues, and I expect that this project will be no exception. It will be my responsibility to fix problems when they appear and I am fully committed to doing so.

I will continue being part of the community, lurking on IRC, willing to share my knowledge and experience, just how my mentors have done with me so far. And of course, since the trash will always need a king™, I also plan on continuing with my contributions to Nautilus and GNOME.

- **What will be showable two months into the project?**

I plan to have internal compression handling and the automatic decompression feature ready after two months.

- **What are your past experiences with the open source world as a user and as a contributor?**

My open-source adventure started when I first installed a **Linux** distribution, soon after I began studying at my university. As a user I went from **GNOME** to **KDE** and then back to **GNOME**, while also playing with **Ubuntu**, **Kubuntu**, **Linux Mint**, **Debian**, **CentOS** and, for the longest period of time, **Fedora**.

As a contributor, my experience began with **Community and Development Lab**, a series of workshops and development sessions which served as an introduction to open-source projects and tools. I worked on a quiz game platform, learning what to expect from an actual project, how to work with version control and how to integrate in and work alongside a community. This year I will be part of the organizing team myself, helping students have their first contact with open-source.

I also joined the organization that held these sessions, **ROSEdu - Romanian Open Source Education**, and had a great experience being a mentor at hackathons and other various events where I got the opportunity to introduce people to open-source.

Other than this, last summer I worked on open-source projects for the **EEA - European Environment Agency**. This is where I gained more technical experience, by being able to work alongside senior developers from whom I had a lot to learn. The highlight of these projects was deploying software using **Docker** containers written entirely by me, such as load balancers, caching and database servers - and seeing them work together so smoothly!

Besides personal work, I encourage the people around me to join open-source communities and try to help them contribute to projects. The help I got when I started contributing was significant to me, so this is how I am trying to “give back to the community”.

- **Please include a link to the committed code or to Bugzilla for the bugs you fixed for the GNOME module your proposal is related to.**

I was was more intrigued by the internal aspects of **Nautilus**, so most of my patches to it revolve around this area. Notable ones are changing the multi-threading API and fixes for file operation mechanisms. I spent quite a lot of time working on the trashing operation, which also gained me the unfortunate title of *king of the Trash*™. A full list of my contributions can be found at :

https://git.gnome.org/browse/nautilus/log/?qt=author&q=razvan

- **If available, please include links to any other code you wrote for GNOME or other open source projects.**

My first contact with **GNOME** as a contributor didn't come through Nautilus, but through a game, **gnome-nibbles**. It was a good place to learn how the **GNOME** ecosystem works. My contributions to it can be found at:

https://git.gnome.org/browse/gnome-nibbles/log/?qt=author&q=razvan

- **What other relevant projects have you worked on previously and what knowledge you gained from working on them?**

Apart from my work on **GNOME** apps, I've contributed to various other open-source projects. A notable one is **World of USO**, a quiz game platform used as support for Operating Systems courses. The platform has also been featured in a blog post on **linux.com**.

World of USO taught me the "basics" of open-source - how to work in a community, how to use version control and, what I found very important at the time, what to expect from a real project.

The platform is implemented in **Python** and **Django** - most of my knowledge of the two comes from working on this project. While designing a new module for the game, I learned how the MVC pattern works, how to use various APIs and, quite essential to me, to always look if something that I need to write does not exist already.

Meetings Registration Tool - a lightweight tool for setting up meetings and handling registrations. The project is written in **Python** using **Flask** and it was a great way to expand my knowledge about the programming language. The most interesting task was to modify how various localization data is displayed, which had me dig my way through a few layers of libraries. In the end it proved to be a great exercise, one that helped me develop skills that I would later use when contributing to **Nautilus**. My contributions to it can be found **here**.

eea.docker.haproxy and **eea.docker.varnish** - configurable **Docker** images. Working with these two meant having to adapt to technologies that I hadn't used (or even knew they exist) before - and on top of that, make them work as **Docker** containers. I learned how useful documentation and code examples can be. My contributions can be found **here** and **here**.

- **What other time commitments, such as school work, exams, research, another job, planned vacation, etc., will you have between May 25 and August 21? What are the dates for these commitments and how many hours a week do these commitments take?**

My exams session starts on 28th of May and ends on 17th of June. Considering this semester has a number of challenging courses, I will not be as active during that time . Other than this I have no other time commitments during the summer, so I should have no problems working extra time on the project to make up for the lost time, if any.