Sprint 2 Planning Document

Team YFFS

Team 22
Brian VerVaet, Max Molnar, Sam Spencer, Wyatt Dahlenburg, Zach Kent

Table of Contents

Sprint Overview

Now that our tools have been successfully implemented and basic functionality, we are hoping to build off of this by implementing more features, such as file directories, filename obfuscation, and more robust user permissions. We are no longer only focusing on independent tools, but now we are going more in depth to alter how the files are stored. This should add to both usability and security.

Scrum Master: Brian VerVaet

Meeting Schedule: Tuesdays, Thursdays, and Sundays at 3:00pm

Risks/Challenges: Most of our user stories involve making large changes to our that will impact each of our tools differently. Last sprint, we were each responsible for one tool and they were largely independent. This sprint, there is a larger overlap, which means small changes could break multiple tools, or even he whole project. Frequent meetings and making sure we stay on top of merges will be a necessity.

Current Sprint Detail

User Story #1

- As a user, I would like to be able to, create directories for files.

#	Task Description	Estimated Time	Owner
1	Research how directories are set up in a file system	4	Wyatt Dahlenburg
2	Modify permissions so that folders are recognized differently than regular files	4	Wyatt Dahlenburg & Zachary Kent
3	Update yffs-ls to display only the files within a specified folder	2	Wyatt Dahlenburg & Zachary Kent
4	Update yffs-add and yffs-edit to insert files into a folder	6	Wyatt Dahlenburg & Zachary Kent
5	Update yffs-rm to remove contents recursively from a folder	5	Wyatt Dahlenburg & Zachary Kent

Acceptance Criteria:

Given a folder to insert in the filesystem, it is expected that this folder will be able to store files.

Given a user wants to list the files in a folder, it is expected that only the files within a folder will be output.

Given a user wants to insert a file in a folder, it is expected that the file is put in the correct folder path.

User Story #2

- As an admin, I would like to maintain ownership and permissions of files.

#	Task Description	Estimated Time	Owner
1	Document intended permissions setup (including default permissions)	1	Sam
2	Research os-checking and user checking on each platform	2	Sam
3	Implement ownership checking for all platforms	2	Sam & Zachary
4	Implement yffs-chmod in order to change permissions	4	Zachary
5	Implement yffs-chown in order to change owner	4	Zachary
6	Write shell script to test permissions with different users	3	Sam

Acceptance Criteria:

Every file should have an owner and permissions defined.

Given a new file addition, that file is automatically set with the default permissions.

Given an invocation of yffs-chmod, the read/write permissions bits can be reset.

Given an invocation of yffs-chown, the owner of a file can be reset.

User Story #3

- As an user, I would like my tools to respect the ownership and permissions of files

1	Change all library functions to appropriately check for owner or user	12	Zach
2	Debug and test ownership permissions	3	Zach
3	Change all library functions to appropriately check for read/write permissions	12	Sam
4	Debug and test read/write permissions	3	Sam
5	Make all necessary changes to each tool (yffs-*) to maintain permission cohesion	10	Sam
6	Debug changes to tools	3	Sam

Acceptance Criteria:

Given a user without correct permissions, when they attempt to use yffs tools they should not be allowed to do so.

Given a user with correct permission, when they attempt to use yffs tools they should be allowed to do so.

Given a user without permissions, when they attempt to use yffs tools then there should be appropriate error message.

User Story #4

- As an admin, to define a hash for all filenames to be obfuscated.

#	Task Description	Estimated Time	Owner
1	Research Hash Functions, general encryption	5 hours	Max
2	Write an encryption function that works with every tool	7 hours	Max
3	Write a decryption function that works with every tool	7 hours	Max
4	Give users a choice of encryption type	6 hours	Max
5	Test encryption/decryption, verify they work correctly	5 hours	Max

Acceptance Criteria:

Given a filename to add, when the user adds it with the encryption flag the file name should be obfuscated.

Given an encrypted filename to read, when the user reads it with the decryption flag, the file name should be unobfuscated and readable.

Given a filename to encrypt, when the user chooses one of the listed encryption options, they should all successfully obfuscate the file names differently.

User Story #5

- As a user, I would like to have a complete documentation of the filesystem tools.

#	Task Description	Estimated Time	Owner
1	Research man and the markup language used	5	Brian
2	Edit makefile to add documentation to appropriate directories	3	Brian
3	Create manpage for yffs library	4	Brian
4	Create manpage for yffs tools: ● yffs-add ● yffs-edit ● yffs-create ● yffs-ls ● yffs-rm ● yffs-cat ● yffs-chmod ● yffs-chown	24	Brian & Zachary

Acceptance Criteria:

Given the yffs library, when the user calls man, a standard linux man page should be output giving an overview of the yffs project, introducing each tool, and pointing the end user to up-to-date source code.

Given a yffs tool, when the user calls man, a standard linux man page should be output explaining the tool and the scope of each tool's functionality.

Given a yffs tool, when the user calls man, a standard man page should be output detailing complete usage instructions with basic examples.

Remaining Backlog

Functional Requirements

- As a user, I would like to be able to,
  - use this file system on embedded devices.
  - Add multiple files at once
  - Redirect ls output

- As a developer, I would like a(n)
  - API to assist with the inclusion of encryption plug-ins.

- As an admin, I would like
  - to define encryption operations on the filesystem.
  - to be able to test the security of encryptions

Non-Functional Requirements

- We must be able to implement this on many different devices.
- The encryption interface should work quickly.
- The system will run efficiently on embedded devices.
- The encryption interface should be easy to use.