DVCS Development
(Group Work)

CSC 253/453
University of Rochester

October 29, 2024

1 DVCS Development and Documentation (200 points

In next three weeks, starting 10/28, 11/4, 11/11, each group is to work in parallel and finish the implementation and documentation of the DVCS system. The deliverables are required as follows (with the required artifacts in bold fonts):

- *Design and management artifacts*
  - **Module Design and Specification (MDS)** You have already submitted the Module Design and Specification (MDS). Document all design changes as comments to the MDS file. The MDS document is append only. You may only augment but not alter what is already in the document and its appendix. Only make a design change in a module if (1) the change is necessary, and (2) it is approved by the person assigned to the module. MDS must be kept consistent with the implementation.
  - **Software Project Management Plan (SPMP)** Submit SPMP at the start of each week (no later than Wednesday, 10/30, 11/6, 11/13, respectively) to include the work plan by every member of the group in the following efforts: design, implementation, testing, and documentation. In the first SPMP, describe the team organization. In the two subsequent SPMPs, summarize the work in the previous week wrt the plan in the previous SPMP. Delays are common and will not cause anyone to lose points, but they should be documented.
  - **Summary of Work** At the end of the project, summarize the efforts by each group member in design, implementation, testing, and documentation.
- *Implementation artifacts* The implementation language is Rust. The platform the DVCS system must support are CSUG machines.

  - **Prototype DVCS** including the prototype code and a document describing its modules and their USE relationship. The prototype is due at the end of the first week, 11/3. The module design and USE graph should be shared with the class on the Discussion Forum.
  - **DVCS** Complete implementation of the DVCS system.

- *Test artifacts* **Unit tests** and **Module tests** should be written in Rust. The **Acceptance Tests** can be written in a scripting language of your choice.

1.1 Prototyping and Demo (20 + 20 points)

Each group is required to develop a prototype and demonstrate in class. Loosely defined, a prototype is an executable that accomplishes some user functions of the DVCS system. The purpose is to demonstrate some aspect to a user. Examples are a concept, e.g. a revision, an interface, e.g. the command line, or capability, e.g. data store and retrieval. It should be presented in the language and operations understandable and expected by an end user.

The Demo is in class on Monday November 4. The requirements are as follows:

- Your group may choose the demonstrate a single prototype involving all three top-level modules.
- By mid-night Sunday November 3, post to the Blackboard Discussions which function(s) will be demonstrated by the prototype. **Each group has up to 10 minutes total time to make the demonstration.**
- The prototype must be compiled and run on a CSUG machine.
- Everyone is required to attend the Demo class. Grading is by attendance (attending the entire class period).
- The demonstration should use the projection screen. You are responsible for bringing and setting up any device you need for the demo. Check the setup the day before to make sure the projection works properly. There is a class before us so you cannot just test before our class starts. Still, you may use the time between classes to double check your setup.
- Set aside at least 3 minutes to allow the audience to suggest test inputs to the prototype. Be prepared to run your prototype with inputs suggested from the audience.
- At each demo, every group not presenting should suggest at least one input for the demo group.

The grading of the demo is based on participation (20 points). The grading of the prototype is based on having at least **one functionality that requires**

all three top-level modules and the clarity of the presentation (4 points) and the quality of the prototypes from a DVCS user's perspective (16 points). The quality includes robustness, which means the system supports a variety of uses and if fails, fails gracefully (again judged from a DVCS user's perspective). Furthermore, note that the clarity of presentation and good time management are pre-requisites for the demonstration of quality. You should go through the demo at least once speaking it aloud before giving the demo in class.

1.2 Weekly Report

At the end of each week, i.e. 11/3, 11/10, 11/17, submit a weekly status report in a CSV file. Each row reports the status of one leaf module, with the following fields.

1. name(s) of the group member contributing to its implementation. If more than one person contributes, include a rough estimate of the portion of contribution.
2. module ID
3. module name, including the name(s) of its parent modules
4. the number of structs, enums, traits in the implementation
5. the total number of public methods and functions in all structs, enums, and traits in the implementation
6. the total number of private methods and functions in the implementation

In addition, each group should add to the Group Design post on the Discussion Board an update each week to share a public report on the following aggregate information: the total number of leaf modules, the total number of structs and enums from all modules, the total number of traits, the total number of public and private methods and functions. Compute these totals from your weekly CSV report.

2 Submission

Weekly Reports The full CSV report, itemized by the leaf modules and individual contributions, should be submitted by or before mid-night 11/3, 11/10, and 11/17. To even out the work, group members should take turns to compose and send the report each week. Include in the report the group name. The group CSV summary should be submitted to the Discussion Forum.

Final Deliverables The final documentation includes the final version of Module Design and Specification (MDS) and the DVCS code repository. The MDS document should be in PDF format and updated with all the changes. Submit these on Blackboard.