

Система для автоматического описания способа получения файлов в проекте по действиям пользователя в консоли

Сорока А.В.

Научный руководитель: Сергушичев А.А.

Санкт-Петербургский национальный
исследовательский университет информационных
технологий, механики и оптики

Проблема

- ▶ Развитие биоинформатического анализа
- ▶ Обработка информации при помощи множества программ
- ▶ Узнать историю происхождения файла
- ▶ Воссоздать его

Пример проблемы

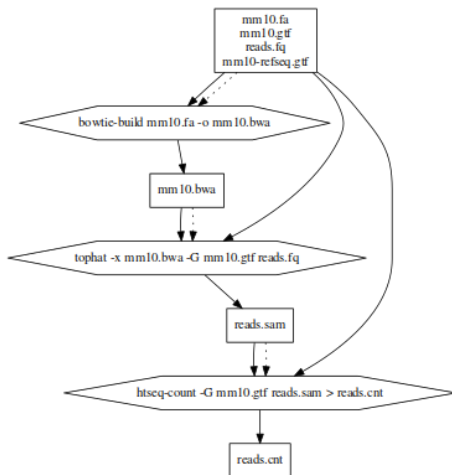


Рис. 1: История сессии

Цель работы

- ▶ Разработать систему автоматического получения истории файла
- ▶ Разработать способ генерации файла на основе его истории

Текущие варианты решения и минусы подходов

- ▶ Использование систем контроля версий
 - ▶ Ручная фиксация
 - ▶ Возможность что-то упустить
 - ▶ Нет информации о зависимостях
- ▶ Makefile
 - ▶ Ручной ввод
 - ▶ Только функция воссоздания
- ▶ Galaxy
 - ▶ Работа в веб-интерфейсе
 - ▶ Инструменты только для биоинформатического анализа

- ▶ Граф истории
- ▶ Вершины – файлы в ревизии
- ▶ Ребра – команда, породившая узел

Реализация

- ▶ unionfs
 - ▶ Возможность объединения нескольких файловых систем друг на друга с выборочным установлением атрибутов на чтение или запись
 - ▶ Возможность cory-on-write в пользовательском процессе
 - ▶ Возможность получения информации о чтении файлов
- ▶ zsh
 - ▶ Возможность автоматического вызова программ после определенных действий пользователя

Состояние

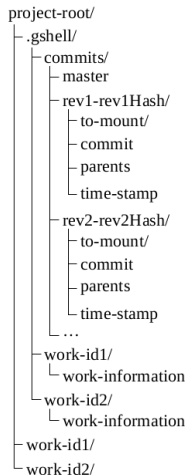


Рис. 2: Состояние

Принцип работы

```
add-zsh-hook preexec preexec_gshell
add-zsh-hook precmd precmd_gshell
preexec_gshell () {
    unset GSHELL_DONE
}
precmd_gshell () {
    ...
    gshell commit "`pwd`" "`fc -n -l -1`"
    cd `pwd`
    GSHELL_DONE=true
    ...
}
```

```
gshell () {  
    ...  
    case $1 in  
        ...  
        enter | enterRev)  
            to_cd=`gShell $@ | tail -1 | awk '{print $2;}'`  
            cd ${to_cd}  
            export GSHELL=true  
        ...  
    esac  
}
```

История сессии

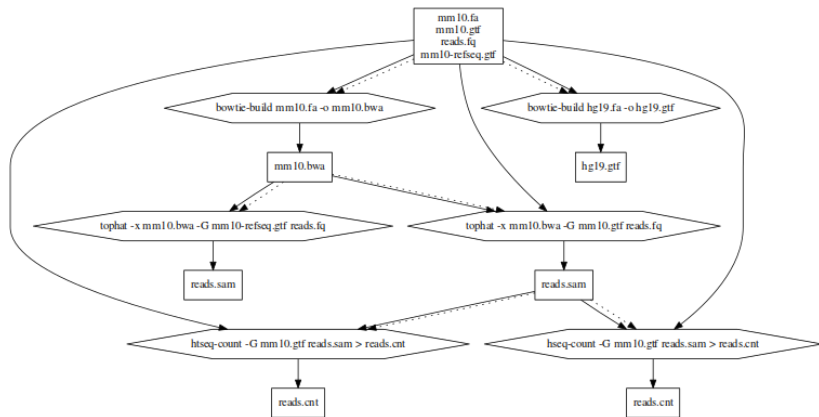


Рис. 3: Граф истории

- ▶ Не требует дополнительных навыков
 - ▶ Используются стандартные средства
- ▶ Автоматизация процесса создания истории
- ▶ Откат действий
- ▶ Параллельная работа
- ▶ Объединение действий

Ограничения

- ▶ Невозможно отследить изменения выполненные в интерактивных процессах
- ▶ Платформ
- ▶ Возможность работы только с unix-like операционными системами

Дальнейшая работа

- ▶ Сборщик мусора для ревизий
- ▶ Реализация в виде виртуальной файловой системы

Заключение

- ▶ Разработан подход для решения задачи автоматического описания способа получения файлов в проекте по действиям пользователя в консоли
- ▶ Подход реализован в виде программы gshell на языке Haskell и zsh

Вопросы?