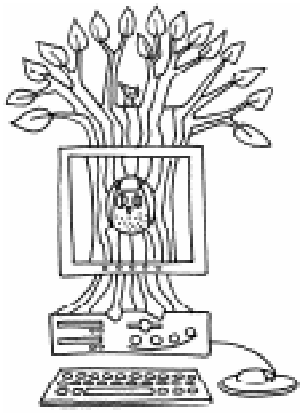
УЧЕБНАЯ МАСТЕРСКАЯ

Колмогоров Константин Алексеевич

РАБОТА С ВЕРСИЯМИ ПРОЕКТА ИЛИ КАК НЕ ИСПОРТИТЬ ЧТО-НИБУДЬ У СЕБЯ В КОМПЬЮТЕРЕ

Тот, кто еще только начинает осваивать компьютер, хорошо знает, да и любой опытный пользователь подтвердит: иногда действительно страшно нажать незнакомую кнопку – совершить незнакомое действие, работая с программой. «Как бы мне чего-нибудь не испортить», – вот фраза, крепко засевшая у нас в голове, когда мы пытаемся применить, к примеру, новый инструмент для рисования в графическом редакторе. А казалось бы, чего бояться: графический файл – это не холст, а инструмент – это не кисть и не краски. Мы не испортим настоящее материальное полотно: виртуальность дает нам право на любую ошибку. Всегда можно вернуться на шаг назад или открыть сохраненный ранее файл.

Это действительно так, поскольку одно из неотъемлемых свойств компьютерной информации – ее неисчерпаемость. С электронными документами легко можно проделать фокус из старого анекдота: «Где Вы их берете? – В тумбочке. – А кто их туда кладет? – Я». Нам остается только выяснить, что, как, куда, а главное – когда класть, чтобы ничего не потерялось и не испортилось.



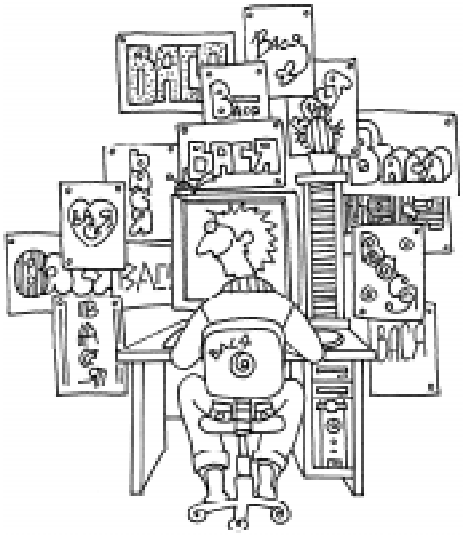
...все файловые системы строятся по... древовидному принципу.

Для начала необходимо разобраться, каким образом наши документы хранятся в компьютере. Ответ таков: они хранятся в виде файлов в файловой системе. На сегодняшний день все файловые системы строятся по так называемому древовидному принципу. Этот принцип подразумевает наличие хотя бы одного так называемого корневого каталога (директории, папки) и файлов в нем. Таким образом, в каталоге могут храниться как файлы, так и каталоги. Условно, файл – это книга, а каталог – книжная полка. В принципе, каталог – это тоже файл, только иной природы. Но не будем вдаваться в подробности о природе файлов, которые, по правде говоря, тоже могут быть каталогами. Остановимся на книгах и книжных полках. Так, у каждого файла и каталога есть имя. У каждого файла и каталога есть путь к нему, он называется полным именем файла.

Опытный компьютерный пользователь, читающий статью, наверное, уже негодует: «Зачем объяснять прописные истины?» Однако данные обстоятельства являются основополагающими и их необходимо себе четко представлять, работая с компьютером. Перейдем к основному предмету нашей статьи и поговорим об этапах создания компьютерного проекта. Под компьютерным проектом можно понимать раз-

рабатываемый программный комплекс, графический файл с логотипом или мультфильм. В конце концов, текстовый документ – это тоже компьютерный проект. Как показывает практика, зачастую создатели очень неаккуратно относятся к так называемым *версиям* своего проекта на различных этапах. То есть часто хранится лишь рабочая версия и предыдущая. Некоторые же самые неаккуратные работают прямо с единственной рабочей версией проекта.

А представьте, что Вы дизайнер. Вы нарисовали логотип, распечатали, повесили на стену. Затем много раз его изменяли и перерисовывали, так что осталась только одна последняя версия. И тут к Вам приходит заказчик, показывает на стену и говорит: «Вот то, что мне нужно!» В итоге, вместо того, чтобы просто открыть нужную версию логотипа, Вам приходится рисовать ее заново. Аналогичную ситуацию может представить себе и программист, и любой другой компьютерный пользователь.



Вы нарисовали логотип, распечатали, повесили на стену.

Итак, постулируем: компьютерную информацию очень легко растиражировать. Процесс получения копии файла не требует от нас практически никаких усилий, чем, собственно, и пользуются пресловутые пираты. Но если пираты пользуются данной особенностью со злым умыслом, нанося тем самым вред владельцам, то создатели электронного проекта могут пользоваться этой возможностью исключительно ради собственной пользы. Здесь важно понимать, что компьютерные проекты при их создании далеко не всегда «развиваются» по возрастающей. Часто, например, при разработке программного обеспечения, программисты вынуждены работать с целыми ветками различных версий проекта и нередко возвращаться к предыдущим отлаженным рабочим версиям, оперировать частями этих версий или выбирать наиболее удачную реализацию.

Существует целый класс программных продуктов, позволяющих работать с целыми системами версий. Разновидностью таких программ является CVS (Concurrent Version System – система управления версиями). Своим происхождением CVS обязана операционной системе Unix, однако существует ее версия и для Windows. CVS предназначена для работы над проектами с открытым исходным кодом, то есть в основном рассчитана на нужды программистов. CVS поддерживает историю дерева каталогов, в которых хранится исходный код, работая с последовательностью изменений. CVS маркирует каждое изменение моментом времени, когда оно было сделано, и именем пользователя, совершившего изменение. Обычно человек, совершивший изменение, также предоставляет текстовое описание причины, по которой произошло изменение. Вооружившись этой информацией, CVS может отвечать на такие вопросы, как

- Кто совершил данное изменение?
- Когда его совершили?
- Зачем это сделали?
- Какие еще изменения произошли в то же самое время?

Таким образом, программисты, применяющие в своей работе CVS, получают больше возможностей контролировать процесс разработки, так как в любой момент времени они могут отследить всю историю внесения изменений в код программы, будь то исправление ошибки или добавление новой функциональной части. У них есть возможность производить над своей программой любые эксперименты, так как они уверены, что всегда смогут вернуться к работающей версии.

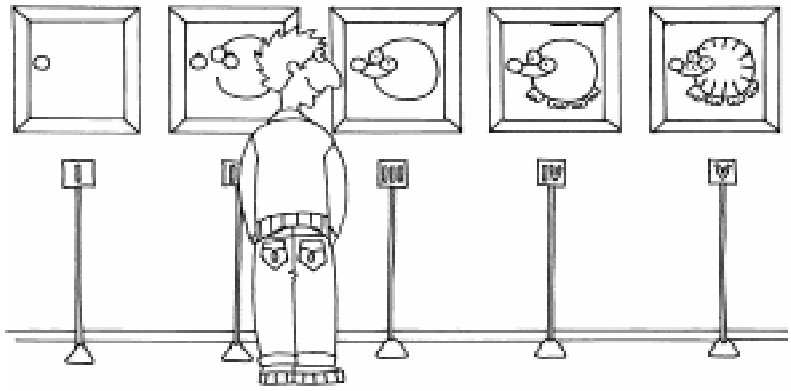
CVS, безусловно, является удобным механизмом при работе над проектом. Однако, к сожалению, он, во-первых, не настолько универсален, чтобы его могли использовать не только программисты, во-вторых, работа с ним требует достаточно высокой квалификации пользователя.

В качестве рекомендаций для рядового пользователя можно предложить следующие основные моменты, упрощенно реализующие функции CVS:

- При работе над проектом использовать для хранения версий каталоги с соответствующими именами.
- Работать только с рабочей версией, помещенной в специальный каталог.
- При внесении любого существенного изменения отдельно сохранять старую версию без последнего изменения в соответствующем каталоге.
- Именовать каталоги с версиями только смысловым содержанием.
- Аналогично именовать все рабочие файлы (документы) проекта.
- Крайне желательно в каталоге с каждой версией хранить метаинформацию, отражающую особенности данной версии.

Следуя данным рекомендациям или, если хотите, просто добрым советам, можно действительно обезопасить себя от возможных неприятных последствий, когда очередное совершенное Вами незнакомое действие (кнопка, опция, пункт меню и т. д.) может что-то испортить в Вашем «хорошем» документе. Более того, поступая при работе подобным образом, Вы можете не бояться экспериментировать как с развитием проекта – созданием новых его частей, так и просто с различными возможностями, предлагаемыми инструментальными средствами. А, как известно, свобода действий дает возможность творить.

В заключение хочется дать еще один добрый совет: перед тем, как назвать свой документ именем, не вполне соответствующим его содержанию (MyPage1.html, Serega.doc, aaa.cpp и т. д.), а также поместить его в каталог с несоответствующим названием, подумайте, что через определенное время Вам вновь может потребоваться этот файл. Насколько же Вам будет приятно с легкостью обнаружить свой файл, который правильно назван и находится в нужной директории.



...в любой момент времени они могут отследить всю историю внесения изменений в код программы...

Литература

1. Домашняя страница CVS: http://www.cvshome.org/
2. Крупнейший ресурс по CVS на русском языке: http://alexm.here.ru/cvs-ru/
3. *Александр Гнатуш.* «Case-технологии: что, где, когда?» IT Manager № 4(16), 2004.
4. *Брукс Ф.* Как проектируются и создаются программные комплексы. М.: Наука, 1979.

Колмогоров Константин Алексеевич, программист, магистр техники и технологий, аспирант СПбГУ ИТМО.