

ВИКИПЕДИЯ

Философия Unix

Материал из Википедии — свободной энциклопедии

Философия Unix — набор культурных норм и философских подходов к разработке программного обеспечения, основанных на опыте ведущих разработчиков операционной системы Unix.

Содержание

Макилрой: четверть века Unix

Майк Ганцарз: философия Unix

Реймонд: Искусство программирования в Unix

Цитаты

Критика

The UNIX-HATERS Handbook

Примечания

Ссылки

Макилрой: четверть века Unix

Дуг Макилрой, изобретатель каналов Unix и один из основателей традиции Unix, обобщил философию следующим образом:

«Философия Unix гласит:

Пишите программы, которые делают что-то одно и делают это хорошо.

Пишите программы, которые бы работали вместе.

Пишите программы, которые бы поддерживали текстовые потоки, поскольку это универсальный интерфейс».

Обычно эти высказывания сводятся к одному «Делайте что-то одно, но делайте это хорошо».

Из этих трёх принципов только третий является специфичным для Unix, хотя разработчики Unix чаще других акцентируют внимание на всех трёх принципах.

Майк Ганцарз: философия Unix

В 1994 году Майк Ганцарз (англ. *Mike Gancarz*) объединил свой опыт работы в Unix (он является членом команды по разработке системы X Window System) с высказываниями из прений, в которых он участвовал со своими приятелями программистами и людьми из других областей деятельности, так или иначе зависящих от Unix, для создания *Философии Unix*, которая сводится к 9 основным принципам:

1. *Красиво — небольшое.*
2. *Пусть каждая программа делает что-то одно, но хорошо.*
3. *Стройте прототип программы как можно раньше.*
4. *Предпочитайте переносимость эффективности.*
5. *Храните данные в простых текстовых файлах.*
6. *Извлекайте пользу из уже существующих программных решений.*
7. *Используйте скриптовые языки для уменьшения трудозатрат и улучшения переносимости.*
8. *Избегайте пользовательских интерфейсов, ограничивающих возможности пользователя по взаимодействию с системой.*
9. *Делайте каждую программу «фильтром».*

Менее важные 10 принципов не снискали всеобщего признания в качестве частей философии Unix, и в некоторых случаях являлись предметом горячих споров (монолитное ядро против микроядра):

1. *Позвольте пользователю настраивать окружение.*
2. *Делайте ядра операционной системы маленькими и легковесными.*
3. *Используйте нижний регистр и придерживайтесь кратких названий.*
4. *Не храните тексты программ в виде распечаток («Спасите деревья!»).*
5. *Не сообщайте пользователю об очевидном («Молчание — золото»).*
6. *Разбивайте сложные задачи на несколько простых, выполняемых параллельно («Мыслите „параллельно“»).*
7. *Объединённые части целого есть нечто большее, чем просто их сумма.*
8. *Ищите 90-процентное решение.*
9. *Если можно не добавлять новую функциональность, не добавляйте её («Чем хуже, тем лучше»).*
10. *Мыслите иерархически.*

Реймонд: Искусство программирования в Unix

Эрик Рэймонд (англ. *Eric S. Raymond*) в книге «Искусство программирования в Unix» подытожил философию Unix как широко используемую инженерную философию «Делай это проще, глупец» (принцип KISS). Затем он описал, как эта обобщённая философия применима в качестве культурных норм Unix. И это несмотря на то, что несложно найти несколько нарушений в следующей текущей философии Unix:

- **Правило модульности:** Пишите простые части, соединяемые понятными интерфейсами.
- **Правило ясности:** Ясность лучше заумности.
- **Правило композиции:** Разрабатывайте программы так, чтобы их можно было соединить с другими программами.
- **Правило разделения:** Отделяйте правила (*policy*) от механизма (*mechanism*); отделяйте интерфейс от движка (*engine*).
- **Правило простоты:** Нацельтесь на простоту; добавляйте сложность, только где необходимо.
- **Правило экономности:** Пишите большую программу только когда другими средствами выполнить необходимую задачу не удастся.
- **Правило прозрачности:** Разрабатывайте прозрачные программы для облегчения последующего пересмотра и отладки.
- **Правило надёжности:** Надёжность — дитя прозрачности и простоты.
- **Правило представления:** Храните знания в данных так, чтобы логика программы была тупой и надёжной.
- **Правило наименьшего удивления:** При разработке интерфейса всегда делайте как можно меньше неожиданных вещей.
- **Правило тишины:** Если программе нечего сказать, пусть лучше молчит.
- **Правило восстановления:** Если надо выйти из строя, делайте это шумно и как можно быстрее.
- **Правило экономии:** Время программиста дорого; сократите его, используя машинное время.
- **Правило генерации:** Избегайте ручного набора кода; при любом удобном случае пишите программы, которые бы писали программы.
- **Правило оптимизации:** Сначала — опытный образец, потом — «причесывание». Добейтесь стабильной работы, только потом оптимизируйте.
- **Правило многообразия:** Отвергайте все утверждения о «единственно правильном пути».
- **Правило расширяемости:** Разрабатывайте для будущего. Оно наступит быстрее, чем вы думаете.

Большинство из этих норм принимается вне сообщества Unix — даже если это было не так во времена, когда они впервые были применены в Unix, то впоследствии это стало так. К тому же много правил не являются уникальными или оригинальными для сообщества Unix. Тем не менее, приверженцы программирования в Unix склоняются к тому, чтобы принять сочетание этих идей в качестве основ для стиля Unix.

Цитаты

- «*Unix прост. Но надо быть гением, чтобы понять его простоту*» — Деннис Ритчи.

- «*Unix не предназначен для ограждения своих пользователей от глупостей, поскольку это оградило бы их и от умных вещей*» — Дуг Гвин.
- «*Unix никогда не говорит „пожалуйста“*» — Роб Пайк.

Критика

The UNIX-HATERS Handbook

Философия UNIX критиковалась в книге «The UNIX-HATERS Handbook», изданной в начале 1990-х годов.

- По мнению редакторов книги, подход Unix приводит к появлению решений, сделанных наспех, без должного продумывания архитектуры, после чего данные решения канонизируются (enshrined), то есть объявляются вечной классикой. Например, таким решением, по их мнению, являются lock files — временные файлы без содержимого, создаваемые как пометка того факта, что какая-то программа находится в процессе исполнения.
- X Window System была подвергнута критике за отделение в ней механизма (engine) от политики (policy), что привело к отсутствию в UNIX стандарта на политики управления пользовательским интерфейсом и большим затруднениям при разработке приложений, использующих GUI.
- NFS была подвергнута критике за изначально порочный подход к архитектуре — попытку создать stateless файл-сервер при том, что это принципиально невозможно. Когда же невозможность поддержки некоторых важных вещей стала очевидной, к NFS прикрутили «костыль» под названием процесса lockd.

Но, в то же время, критикуемые в этой книге подходы, начатые в *NIX, плавно обосновываются и в ОС Microsoft Windows и Apple Mac OS.

Примечания

Ссылки

- Brian Kernighan, Rob Pike. The UNIX Programming Environment (<http://cm.bell-labs.com/cm/cs/upe/>) 1984
- Rob Pike. Notes on Programming in C (<http://www.lysator.liu.se/c/pikestyle.html>) 11 сентября 1989
- Richard Gabriel. The Rise of Worse is Better (http://xahlee.org/UnixResource_dir/fastfood_dir/worse-is-better.html) // Lisp: Good News, Bad News, How to Win Big (<https://web.archive.org/web/20060913023048/http://www.ai.mit.edu/docs/articles/good-news/good-news.html>) 1991
- Peter H. Salus. A Quarter Century of UNIX. Addison-Wesley, 31 мая 1994. ISBN 0-201-54777-5

- Eric S. Raymond. Philosophy (<https://web.archive.org/web/20080512141528/http://www.faqs.org/docs/artu/philosophychapter.html>) // The Art of UNIX Programming (<http://www.catb.org/~esr/writings/taoup>). Addison-Wesley, 17 сентября 2003. ISBN 0-13-142901-9
 - M. D. Schroeder, D. D. Clark, J. H. Saltzer, D. H. Wells. Final Report of the Multics Kernel Design Project (<http://citeseer.ist.psu.edu/sc Schroeder77final.html>) 1977
 - *Joel Spolsky*. Biculturalism (<http://www.joelonsoftware.com/articles/Biculturalism.html>) (англ.). *Joel on Software* (14 December 2003). — Взгляд Windows-разработчика на различия двух культур. Архивировано (<http://www.webcitation.org/65E5Vn2JZ>) 5 февраля 2012 года.
 - *Денис Смирнов*. Классический UNIX-way или «компьютер для профессионала» (http://freesource.info/wiki/Stat'ja_Klassicheskijj_Unix_Way) (2004). — Разъяснение базовых принципов Unix простым языком.
-

Источник — https://ru.wikipedia.org/w/index.php?title=Философия_Unix&oldid=106825688

Эта страница в последний раз была отредактирована 5 мая 2020 в 20:53.

Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.
Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.