

История развития UNIX систем.

Период 1968-69 гг. был очень неопределенным для Bell Labs: проект операционной системы с разделением времени Multics (Multiplexed Information and Computing Service), разрабатываемой с 1964 года для дорогой 36-битной ЭВМ GE-645, не имел четких перспектив и целей, а лишь разрастался в размерах и сложности, всё ясней был виден его предполагаемый провал. В конце концов, American Telephone & Telegraph вышла из проекта, в который за пять лет были вложены миллионы долларов.

Однако, некоторые из инженеров, работавших над Multics — Кен Томпсон, Деннис Ритчи, Малкольм Дуглас Макилрой, Джозеф Оссанна, — ощущали необходимость в продолжении работы над подобным проектом, и не хотели терять успешную сформировавшуюся уютную рабочую атмосферу. Поэтому в 1969 году они начали искать альтернативу Multics: Оссанна, Томпсон, Ритчи пытались протолкнуть покупку машины средней мощности, для которой группа обещала написать операционную систему. Заказ на предлагаемые ЭВМ DEC PDP-10 и Sigma 7 так и не был сделан, и хотя несколько раз ситуация выходила на грань получения нужного оборудования, было вполне очевидно, что команда просит слишком крупную сумму для проекта с расплывчатым планом, тем более, что после провала Multics операционные системы стали не настолько привлекательной сферой. Поэтому Томпсон (в основном это была его задумка), Кэнэдэй и Ритчи (привнесший идею файлов-устройств) разработали на обычных черных учебных досках и бумаге устройство файловой системы, которая позже стала сердцем Unix.

Начало работы

Томпсон написал для Multics симуляцию производительности предлагаемой файловой системы и страничного поведения программ, начал реализацию новой операционной системы для GE-645. Он разработал ассемблер для машины и рудиментарное системное ядро, способное лишь выводить приветственное сообщение. Сложность оборудования делала заметным достижение Томпсона, но как только выяснилось, что ввиду выхода из проекта Multics 645-ая серия в Bell Labs

будет продолжать работать от силы несколько месяцев, работа была прекращена.

Также в 1969 году Томпсон написал игру Space Travel для Multics, пока ещё имел к ней доступ, а потом переписал её на Фортране для GECOS, операционной системы машин GE-635. Игра содержала всего лишь симуляцию движения основных тел Солнечной системы, среди которых игроку предлагалось провести корабль и приземлиться на планетах и их лунах. Игра в версии для GECOS имела два основных недостатка: высокая цена вычислительного времени (\$75 за игру) и подтормаживания графики. Поискав, Кен нашел немного использовавшийся PDP-7 с отличным экраном, ранее машину купили как терминал Graphic-II для проекта анализа цепей.

Ритчи и Томпсон переписали Space Travel для запуска на этой ЭВМ. Работа над маленькой игрой была куда более важной, чем может показаться: поскольку они проигнорировали существующее программное обеспечение, приходилось писать пакет для обработки чисел с плавающей запятой, спецификации графических символов и систему отладки самостоятельно. Всё это было написано на GECOS на кросс-ассемблере для PDP-7, и процесс переноса ассемблерного кода на перфокартах использовался в дальнейшем для реализации уже файловой системы, проект которой уже был разработан.

Летом 1969 года жена Кена уехала к его родителям, чтобы показать новорожденного сына, и Томпсон с головой ушел в работу. Вместе с файловой системой новой операционной системы появился маленький набор пользовательских программ: копирование, печать, удаление, редактирование и, конечно же, интерпретатор команд, командный процессор. После доработки ассемблера отпала необходимость писать на GECOS и переносить файлы перфолентой. Хотя название Unix как отсылка к Multics появилось только в 1970 году как шутка Брайана Кернигана на тему того, что новая система поддерживала лишь пользователя Томпсона, и поэтому должна носить название Un-multiplexed Information and Computing Service, операционная система уже родилась и жила.

Файловая система Unix для PDP-7 имела линейный массив индексных дескрипторов, каждый из которых описывал файл, директории (особый файл, содержащий имена и ассоциированный индекс), особые файлы, описывающие устройства. Вызовы файловой системы тоже не слишком отличаются от сегодняшних: read, write, open, creat, close. Отличие было в том, что единицей ввода-вывода в PDP-7 было машинное слово, а не байт. Прimitивная по сегодняшним меркам операционная система предоставляла более удобную среду разработки, чем существовавшие аналоги. Всё очевиднее становилось, что изначально устаревшая PDP-7, накладывающая множество раздражающих ограничений на работу новой операционной системы, не сможет удовлетворить нужды для дальнейшего развития, как и вся серия машин PDP-7, а разрешать закупку нового оборудования руководство вряд ли решится.

Оссанна, до конца его дней увлекавшийся текстовым процессингом, предложил схитрить и попросить у начальства новейший микрокомпьютер компании DEC — PDP-11 — для создания инструментов редактирования и форматирования текста. А факт того, что для создания этих инструментов потребуется написать операционную систему, шел почти что как сноска. Запрашиваемая сумма была куда меньше, чем ранее — всего 65 тыс. долларов.

Предыдущие расплывчатые предложения встречались с неодобрением, это же было слишком узко. Не сразу, но начальство клюнуло, уже в мае 1970 года благодаря усилиям Макмахона был произведен заказ на PDP-11. Машина прибыла в конце лета, но продукт был настолько новым, что накопительный диск пришлось ждать до декабря. Всё это время Томпсон, Ритчи и остальные продолжали работать над Unix на PDP-7. В это время новенький PDP-11 три месяца обсчитывал ходы шахматного коня на доске 6×8.

Лишь после установки накопителя PDP-11 всё усложняющаяся операционная система была переписана на новую машину. У версии Unix для PDP-11 не имелось особых преимуществ относительно версии для PDP-7. К примеру, всё так же не было многозадачности: лишь одна пользовательская программа могла находиться в ядре. Размеры системы

были невелики: 24 килобайта ОЗУ (16 килобайт для операционной системы и 8 килобайт для пользовательских программ), диск с тысячей блоков (512 килобайт). Файлы были ограничены размером в 64 килобайта.

Затем была переписана программа roff, текстовый процессор, написанный Осанной. Впервые Unix испытали вне Bell Labs при работе в патентном отделе AT&T, где требовалось редактировать и форматировать заявки на патенты. Unix-система имела преимущества над испытываемой в это же время в патентном отделе коммерческой системой редактирования текста: во-первых, Unix имела поддержку 37-ой модели терминалов Teletype, печатавшей все требующиеся математические символы, во-вторых, в roff быстро добавили возможность нумеровать страницы, чем конкурент не обладал.

Во время второго полугодия 1971 года группа осуществляла поддержку трех машинисток и пыталась продолжать работу в свободное время. Unix получила хорошую репутацию за поддержку интересных функций на скромном оборудовании, и система была принята к использованию в патентном отделе AT&T, что давало возможность группе убедить руководство заказать ещё одну машину, другую модель PDP-11 — PDP-11/45, ещё более мощную, чем уже купленная.

В ранние годы Unix постоянно изменялась и эволюционировала, поэтому не было смысла выпускать версии или релизы, хотя периодически происходили издания программистского руководства, и первые системы Unix носили названия этих изданий. Первое издание датируется ноябрем 1971 года.

Языки программирования высокого уровня

Каждая программа для Unix в версии PDP-7 писалась на очень слабом ассемблере — к примеру, не было макрокоманд. Первым интересным языком был TMG Макклера, реализованный Макилроем. Потом Томпсон решил, что нельзя предлагать компьютерную систему без Фортрана, поэтому он начал писать Фортран в TMG, но намерение реализовать его продлилось неделю. То, что он сделал, было компилятором для нового языка B. Новый язык испытал сильное влияние BCPL, вкуса Томпсона к

спартанскому синтаксису и скромные размеры места для компилятора, который производил простой и медленный, но исполняемый код, что сделало жизнь программистов легче. Позже кросс-компилятор PDP-7 для PDP-11 был написан на В, и постепенно компилятор для PDP-7 был транслитерирован из TMG в В. С прибытием PDP-11 почти сразу же на ней был настроен В. Ещё до прибытия накопителя для новой машины на PDP-11 был написан калькулятор.

Над переписыванием операционной системы на В не особо задумывались, даже ассемблер для новой ЭВМ переписывался на ассемблере, поскольку интерпретируемый код был слишком медленным. Также стояла проблема несоответствия ориентированного на машинное слово В и PDP-11 с байтовой адресацией.

Поэтому в 1971 году была начата работа над тем, что стало языком С. Всем прекрасна известна связь $BCPL \rightarrow B \rightarrow C$, и нет смысла повторять её, стоит лишь отметить, что в 1973 году ядро операционной системы было переписано на С, и Unix приобрел черты своих современных версий. Было ещё множество внешних изменений, и внутреннее строение системы стало более рациональным и простым. Успех попытки подтвердил, что С является полезным и практически универсальным инструментом для системного программирования, а не игрушкой для простых приложений. Сегодня лишь самые важные программы Unix-подобных систем пишутся на ассемблере, почти все утилиты и приложения написаны на С.

Дальнейшее развитие

Годами Unix оставалась лишь исследовательским проектом Bell Labs, но в 1973 году авторы представили систему на симпозиуме Ассоциации по вычислительной технике, и запросы на предоставление копий посыпались сразу же. Это ставило AT&T в крайне неудобное положение: ещё в 1956 году монополист связи был ограничен от продажи продуктов и услуг, не связанных напрямую с телефонами и телекоммуникациями, и поэтому Unix не могла продаваться. Именно этим мы обязаны выпуску исходных кодов ранних версий практически всем желающим без поддержки, рекламы и исправлений.

И такая политика дала свои плоды: поклонники операционной системы уже тогда начали общаться и собираться вместе для обмена опытом, исправлениями и багфиксами, новыми инструментами и улучшениями системы. К популярности Unix в конечном счете привели не только простота, элегантность её устройства, востребованность и портируемость, но и формирование сплоченного комьюнити Usenix. Сообщества формировались не только в США, но и в Австралии (в университете Нового Южного Уэльса и Сиднейском университете), и все они обменивались магнитными лентами с новыми программами и исправлениями для системы. К середине 70-х годов именно это окружение сформировало современные традиции сообщества open-source.

Однако, чем популярнее становилась Unix, тем пристальнее юристы AT&T следили за несколько фривольным поведением лицензиатов. К примеру, Джон Лайонс, преподающий в университете Нового Южного Уэльса, в 1977 году опубликовал одну из самых известных книг всех времен «Комментарии к операционной системе Unix», включавшую и листинг исходных кодов ОС. Условия лицензирования Unix допускали обмен исходниками, поэтому вначале книга Лайона продавалась только лицензиатам. В 1979 году AT&T ужесточила наказание за использование книги в учебных классах, и нет ничего удивительного в том, что книга стихийно начала копироваться самиздатом, порой, фотокопии фотокопий имели практически нечитабельное состояние.

Юристы AT&T следили и за жизнью в Bell Labs. К примеру, между изданием шестого издания Unix в 1975 году и седьмого в 1979 Томпсон собрал из внешних и внутренних источников Bell Labs десятки важных исправлений для системы, которые собирался вбросить в пользовательскую базу. Юристы компании запретили ему это делать, поскольку это было бы формой поддержки. Но корректоры ошибок всё равно стали доступны: Лоу Кацу, основателю и презиенту Usenix, однажды позвонили и сообщили, что если он в 2 часа дня посетит определенное место на Маунтайн авеню (где и находится Bell Labs), то он найдет кое-что интересное. Кац последовал указаниям и нашел магнитную ленту с исправлениями, которые немедленно попали в руки бесчисленного числа пользователей.

80-е. Ответвления

После разрушения монополии Bell System компания AT&T более не была связана путами соглашения 1956 года. Одновременно с этим развивались многочисленные ответвления Unix, в первую очередь версия университета Беркли, которая была тщательно вычищена от кода AT&T, чтобы сделать её свободно распространяемой. После суда в 1992 году AT&T против университета Беркли законный статус Unix-клонов был подтвержден.

Если бы исходные коды Unix не были бы доступны общественности, Линус Торвалдс, как он сам говорит, вряд ли бы создал Linux, который в начале 90-х был написан с нуля. Сегодня именно Linux и семейство BSD несут крест отошедшей на задний план Unix и управляют всем: от мобильных телефонов до суперкомпьютеров.

К концу 70-х Unix распространялась в академических кругах, словно дешевый наркотик. Пик её популярности пришелся на ранние 90-е. По ряду причин она уступила множеству коммерческих и некоммерческих ОС, но её элегантное наследие продолжает активнейшим образом развиваться. Сегодня можно говорить, что Unix была самой популярной и влиятельной операционной системой в истории, сформировавшей вокруг себя мыслящее по-новому сообщество.

Интересные факты

UNIX-время

UNIX-время — система описания моментов во времени, принятая в UNIX и других POSIX-совместимых операционных системах. Определяется как количество секунд, прошедших с полуночи (00:00:00 UTC) 1 января 1970 года (четверг); время с этого момента называют «эрой UNIX».

UNIX-время представлено целым числом, которое увеличивается с каждой прошедшей секундой без необходимости вычислений для определения года, месяца, дня, часа или минуты для удобства восприятия человеком. Современное UNIX-время согласуется с UTC — отсчет происходит в

секундах СИ. Временной промежуток одного дня почти всегда разбит на 86400 секунд, но при объявлении високосных секунд составляет 86401 секунду. Такие секунды, согласно Всемирному времени, сохраняют длительность дней синхронизированной со временем оборота планеты. В UNIX-времени соответствующие номера секунд повторяются, то есть високосные секунды не учитываются.

В момент времени 00:00:00 UTC 1 января 1970 года (четверг) UNIX-время равно нулю. Начиная с этого времени, число возрастает на определённое количество в день. Каждый день число, представляющее Unix-время вычисляется описанным образом в UTC (00:00:00Z), и увеличивается ровно на 1 в секунду, начиная с полночи. Следовательно, 16-09-2004 17:55:43.54, 64543.54 секунд с полуночи, из примера выше, будет представлено в UNIX-времени числом $1095292800 + 64543.54 = 1095357343.54$. Для дат, предшествующих началу отсчета, число также возрастает, т.е. с течением времени приближается к нулю.

Используемую целочисленную систему удобно использовать для сравнения и хранения дат (дата и время в этом формате занимают всего 4 или 8 байт). При необходимости обращения к элементам дат (день, месяц, год) секунды можно преобразовать в любой подходящий формат (и наоборот).

В программах для хранения UNIX-времени используется целочисленный знаковый тип. 32-битные числа со знаком могут ссылаться на моменты времени от пятницы 13 декабря 1901 года 20:45:52 до вторника 19 января 2038 года 03:14:07 включительно.

19 января 2038 года в 03:14:08 по всемирному времени значение переменной типа **time_t**, отсчитывающей число секунд, прошедших с 1 января 1970 года, достигнет 231, что может привести к ошибочной интерпретации этого числа как отрицательного. Возможное решение данной проблемы состоит в использовании для хранения времени не 32-битной, а 64-битной переменной (что и делается во всех современных 64-битных операционных системах), этого хватит на 292 млрд лет.

Проблема Apple iOS как UNIX-системы

У 64-битных iOS устройств от Apple существует проблема как UNIX системы. Если перевести время на устройстве с 64-битным процессором под управлением iOS на час ночи первого января 1970 года и перезагрузить устройство, будучи в часовом поясе от UTC +1:30 и больше, то после перезагрузки устройства оно не будет включаться, всё время будет светиться "белое яблоко". Происходит это из-за разницы в часовых поясах, то есть: если перевести время на 1:00 1 января 1970 года в часовом поясе UTC +1:30 или больше, то счётчик UNIX-time уходит в минус, так как отсчёт ведётся от UTC времени, что система понять не в состоянии, вследствие чего счётчик зависает. Устройство даже не восстанавливается через DFU, но у проблемы есть три решения другими способами. Первый способ: просто оставить включенным устройство и ждать пока счётчик UNIX-time уйдёт в плюс (пока не доказано, что данный способ работает). Второй способ: ждать пока аккумулятор телефона сядет полностью и счётчик сам сбросится на нулевое состояние. Третий способ: разобрать устройство и отключить аккумулятор на некоторое время, затем собрать устройство, счётчик так же сбросится на ноль и устройство будет работать.