

<https://habr.com/ru/post/190082/>

holmuk 14 августа 2013 в 19:59

Краткий очерк истории Lisp машин

Lisp

Привет, Хабр!

В некоторых статьях я замечал, что авторы обещали рассказать про историю Lisp машин, но так и не рассказали. Возможно, эта короткая статья будет интересна любителям истории IT.

Да, на свете было мало машин с аппаратной поддержкой лямбда исчисления — около 7000 штук в мире по состоянию на 1988 год. А сейчас гораздо меньше, конечно же. На самом деле Лисп машины сыграли довольно сильную роль в истории информационных технологий. Благодаря этим странным компьютерам мир получил такие технологии, как лазерная печать и многооконный GUI.

Предыстория

В 1930-х годах некоторые математики пытались осознать и анализировать понятие вычислимости. Одну из таких попыток сделал Алонзо Чёрч, создав λ -исчисление. Писать здесь про него я не буду — слишком большая тема. Предполагается, что вы хотя бы знаете это слово :).

Исчисление исчислением, но надо бы его применить. Когда начались работы в области искусственного интеллекта, возникла потребность в языке программирования, позволявшем адекватно решать задачи в этой области. В это время Джон Маккарти работал в MIT и с 1958 по 1963 года разрабатывал Lisp, взяв за основу IPL, предназначенный для реализации ИИ, способного автоматически выводить теоремы математической логики. Первая версия интерпретатора Lisp работала на IBM 704. В 1960 Lisp был портирован на PDP-1. В общем, Lisp распространение получил. Была только одна проблема — Lisp работал довольно медленно на классическом (то есть стандартной архитектуры в данном контексте) компьютере. Прежде чем перейти к самой истории Lisp машин, желательно поверхностно рассмотреть их архитектуру.

Теория Лисп-машины

Возьмем сферическую реализацию Lisp машины и взглянем на неё. Как мы уже знаем, программируют её на Lisp'e. Конечно же, процессор не исполняет Lisp код напрямую, но микрокод и стэковая машина оптимизированы для исполнения кода. Это позволяло спокойно работать и с рациональными, и с комплексными числами. ОС использует виртуальную память для обеспечения достаточно большого адресного пространства, а сборщик мусора автоматически удаляет ненужные предметы из памяти, освобождая место для новых вещей (ну про это вы точно знаете, суть в том, что сборщик мусора таки реализован). Архитектура машины относится к tagged arch., я сейчас вкратце поясню, что это такое, так как информации про нее на русском крайне мало.

Tagged (помеченная?) архитектура предполагает, что каждое слово в памяти содержит в себе так называемую метку, делящая слово на две части — в первой части содержится сама информация, вторая часть описывает, каким образом её следует интерпретировать, а также CDR код, упрощающий сборку мусора в разы. Хватит теории, перейдем к «железу».

Аппаратные реализации λ -исчисления

Наступил 1973 год. В лаборатории ИИ при Массачусетском институте всерьез задумались над созданием архитектуры компьютера с аппаратной поддержкой LISP, поскольку в те времена исполнение LISP программы на компьютере стандартной архитектуры было занятием не для слабонервных. Типизация одной переменной могла занимать несколько минут, поскольку выполняется она во время исполнения программы, а не её компиляции. Дело создания «чудо-архитектуры» поручили двум программистам — Гринблэтту и Найту.

«Черновая» 24-битная версия архитектуры была названа в честь оператора конструирования списков и называлась CONS. Именно по этой машинке Найт защищал свою докторскую диссертацию. В дальнейшем архитектуру этой машины улучшили и назвали CADR (в честь лисп-функции cadr). Но одно дело разработать архитектуру, а другое дело — создать компьютер. И тут на арену

выходит Symbolics, Inc.

Symbolics, Inc.

Symbolics, Inc. была основана Русселом (Russell Noftsker) в Массачусете, в дальнейшем она переместилась в Калифорнию. Компания занималась производством LISP-машин, а также написанием ПО для них. Она была не единственной компанией, производящей странные компьютеры, кроме неё, этим же делом занималась Lisp Machines, Inc, но на стороне Symbolics было большее количество специалистов и желания. Кстати, symbolics.com был, возможно, первым Интернет доменом в зоне .com.

В 1981 данной компанией была создана первая и последняя 32-битная машина CADR архитектуры — CADR LM-2 с 1МБ RAM на борту. ОС и приложения для нее писались, как ни странно, на самом Lisp'e, а если быть точнее, то на его диалекте Lisp Machine Lisp. В 1982 была создана операционная система Open Genera и портирован Emacs (Zmacs), стало возможным работать на LISP-машине и показывать жалким людишкам всю мощь лямбда-исчисления.

Именно в Open Genera и Lisp Machine OS появились множество вкусных плюшек, таких как сборщики мусора, менеджер виртуальной памяти, многозадачная GUI, отладчик и поддержка Chaosnet.

В результате перепалок между MIT и Symbolics по поводу лицензий (как вы уже поняли, здесь побывал и Ричард Столлман), их сотрудничество закончилось где-то около 1983 года. А Ричард, как ни в чем ни бывало, организовал свою шайку LISP-хакеров, состоящую преимущественно из MIT-людей.

В 1983 Symbolics представила первую модель новой линейки LISP машин — Symbolics 3600, размером с домашний холодильник, основанный на CADR архитектуре. Конечно же, не обошлось и без новшеств. Помимо «инновационного» нового дизайна машинки получили аппаратную поддержку сборщика мусора и виртуальной памяти. Процессор стал 36-битным, память же оперировала 44 битными словами, из них 8 бит служили для коррекции ошибок (ECC). Архитектура 3600 содержала в себе 4096 регистров (CISC, привет!), из которых половина использовалась для стека вызовов, а другая половина — микрокодом и любящими время программами, а так же драйверами (Кстати, насколько я знаю, понятие драйвера было введено в процессе создания Lisp-машин).

3600 был построен на TTL интегральных схемах и имел частоту около 5МГц. Не густо, но с учетом аппаратной поддержки LISP в то время было очень даже густо. Поддержка Chaosnet была прекращена, отныне Symbolics поддерживала только Ethernet. Всего за три года было выпущено пять представителей линейки 36xx — 3600, 3670, 3640, 3675 и 3645. Последующие модели отличались только количеством мегагерц и прочих цифр, особых изменений в архитектуре не было.

Представители линейки 3600 были очень популярны в сфере исследования искусственного интеллекта. Кстати говоря, машины 3600 линейки были способны обрабатывать видео HDTV качества, что и послужило причиной их популярности в Японии.

В 1988 эта же компания представила миру линейку Ivory, первым представителем которого был XL400. Эти же компьютеры были уже 40-битными и позволяли адресовать 16ГБ памяти. Чипы архитектуры Ivory производили VLSI и Hewlett Packard, а использовали их, помимо самой Symbolics, также Sun и Apple (да-да, Lisp-процессор использовался и в Макинтошах, правда, не в качестве основного).

Давайте попробуем взглянуть на ассемблерный исходник для процессоров Ivory:

```
ENTRY: 2 REQUIRED, 0 OPTIONAL
PUSH 0
PUSH FP|3
PUSH NIL
BRANCH 15
SET-TO-CDR-PUSH-CAR FP|5
SET-SP-TO-ADDRESS-SAVE-TOS SP|-1
START-CALL FP|2
PUSH FP|6
FINISH-CALL-1-VALUE
BRANCH-FALSE 15
INCREMENT FP|4
ENDP FP|5
BRANCH-FALSE 6
SET-SP-TO-ADDRESS SP|-2
RETURN-SINGLE-STACK
```

Что равносильно следующему коду на Lisp:

```
(defun example-count (predicate list)
  (let ((count 0))
    (dolist (i list count)
      (when (funcall predicate i)
        (incf count))))))
```

Неплохо, да? Как вы видите, мнемоники были довольно длинными, это даже и ассемблером сложно назвать в привычном моем понимании. Если вам интересна суть ассемблера Ivory и других машин, можете глянуть документы на bitsavers.org. Там же есть много других мануалов Symbolics.

Стоит упомянуть и попытку компании создать RISC процессор на базе Ivory — Sunstone. Но проект был свернут. К LISP машинам также приложили свою руку BBN и Xerox. Творением BBN являлась машина Jericho, которая программировалась на Inerlisp. Данное творение никогда не продавалось. Xerox Parc же в 1980-х создал свою Xerox Lisp Machine, которая известна тем, что имела полноценную среду разработки InterLisp-D и первое Hypertext приложение (NoteCards).

Известны удачные попытки создания LISP машин японскими инженерами, из таких машин можно назвать TAKITAC-7 и EVLIS, разработанный в университете Осака. К сожалению, про них я ничего в данный момент рассказать не могу.

The End

Но ничто не могло остановить все увеличивающуюся мощь классических компьютеров. Аппаратная поддержка лямбда-исчисления стала ненужной. В конечном итоге Symbolics обанкротилась, а к 1995 году закончилась эра LISP машин. Но мы их не забудем, правда же? :)

В заключение хочу сказать, что лисп-машины — это не единственная попытка сделать компьютер с аппаратной поддержкой языка X. Существовали реализации Prolog и Java машин, но об этом как нибудь в другой раз. Спасибо за прочтение :).

P.S. Если увидите неточность или ошибку, сообщите в личку.

Теги: Lisp, Лисп, машина, Symbolics, Чёрч, лямбда, исчисление

Хабы: Lisp