PDP-8. ОС на левой коленке

«То, что мы делаем, уже сделано.
То, что мы собираемся сделать – сделано.
То, что мы представляем, что можно сделать, - сделано.
И даже то, что не представляем, - сделано тоже».
Из изречений Владимира Олеговича Молодцова, доцента каф. ЛЭТИ.

В начале было слово, и слово было 12-разрядным

В 1977 году на кафедре Вычислительной техники НЭТИ в аудитории 225 первого корпуса стояла мини-ЭВМ «Электроника-100И» с заводским порядковым номером 29, выпущенная, кажется, Новосибирским электромеханическим заводом, который выпускал еще и популярную магнитофонную приставку «Нота», но значительно большим тиражом. Стояла она с закрытым кожухом, на котором стояла, как помнится, маленькая пластмассовая пальмочка. Как оказалась, она не была сиротой в компьютерном мире, а клоном первой в истории мини-ЭВМ PDP-8 (на самом деле, не одной модели, а целого семейства). В оригинале она комплектовалась дисковыми и ленточными накопителями и имела собственную дисковую ОС (руководства по работе с которой можно было прочитать в ГПНТБ). У нас все было прозаичней. Электроника-100И была укомплектована электрической печатной машинкой и устройством ввода-вывода с перфолент. Работа на ней выглядела так. Оперативная память на ферритовых кольцах сохраняла свое содержимое при выключении памяти, поэтому в памяти всегда был вторичный загрузчик. Если его не было, то в память вручную в пульта заносился в двоичном коде первичный загрузчик из 10 команд, который при пуске читал в память с перфоленты вторичный загрузчик. С его помощью можно было загрузить редактор текста, транслятор с Ассемблера, отладчик, интерпретатор Бейсик-подобного языка Фокал, вот, практически и все. Результаты трансляции и редактирования также вводились и выводились на перфоленту. При трансляции ассемблерного текста файл читался два раза, соответственно, требовалось два раза вставлять ленту. Еще. 12-разрядное машинное слово, оперативка от 4 до 32 К слов, тактовая частота - 1 Мгц.

На пульте процессора по адресу 1142 команда с кодом 5341 (JMP.-1) - безусловный переход на одну команду назад на текущей странице (128 слов) к команде опроса готовности клавиатуры пишущей машинки

Голая теория

Первое задание, которое я получил, направив свои стопы на кафедру Вычислительной техники в поисках научной работы, была разработка языка с системой динамического распределения памяти. Поскольку единственными реально доступными языками программирования тогда были Ассемблер и Фортран (Си и Паскаль появились как раз тогда, но еще не доползли до наших окраин), я на самом деле имел достаточно смутное представление об этом (точнее о том, зачем это надо в программировании). Очевидно, что моему руководителю, **Александру Антоновичу Малявко**, читавшему нам курс «Системное программирование», тогда мерещилось что-то Си-подобное.

Я с энтузиазмом прочитал книжку Гриса «Конструирование компиляторов для электронно-вычислительных машин». Издание насколько интересное, настолько и странное. Главной фишкой в архитектуре было тогда почти повсеместное отсутствие аппаратного стека в процессорах, поэтому понятие рекурсивной функции не фигурировало в изложении материала. (Кстати, Фортран до сих пор является языком, где не допускаются рекурсивные вызовы).

Затем я начал разрабатывать транслятор, используя полученные мной знания о технологии программирования, т.е. начал рисовать блок-схему программы, развернув рулон миллиметровой бумаги (миллиметровки) на столе. Когда блок-схема доросла до другого конца стола, я окончательно в ней запутался, и энтузиазм мой угас. Тем не менее, некоторые компоненты лексического анализатора я даже написал на Ассемблере и отладил.

По поводу проектирования трансляторов. Только почитав несколько лет подряд курс «Теория языков программирования и методы трансляции», я понял, как это делается, по существу (см. http://ermak.cs.nstu.ru/trans).

К тому времени, идеи моего руководителя, также претерпели изменения. К летней практике он предложил мне, не много ни мало, как разработать операционку для означенного компьютера, ибо имеющийся вариант работы с перфолентами не позволял сделать что-нибудь приемлемое для многопользовательского режима.

Летом 1978 года я прочитал еще одну (или даже две) книжки, на этот раз по операционным системам, ничтоже сумняшися, принялся писать (кодировать) ядро ОС «по книжке» - диспетчеры, мониторы и тому подобное, не осознавая до конца, что же это такое. Естественно, кулек с перфолентами, содержащими этот код, целиком пошел в мусорную корзину.

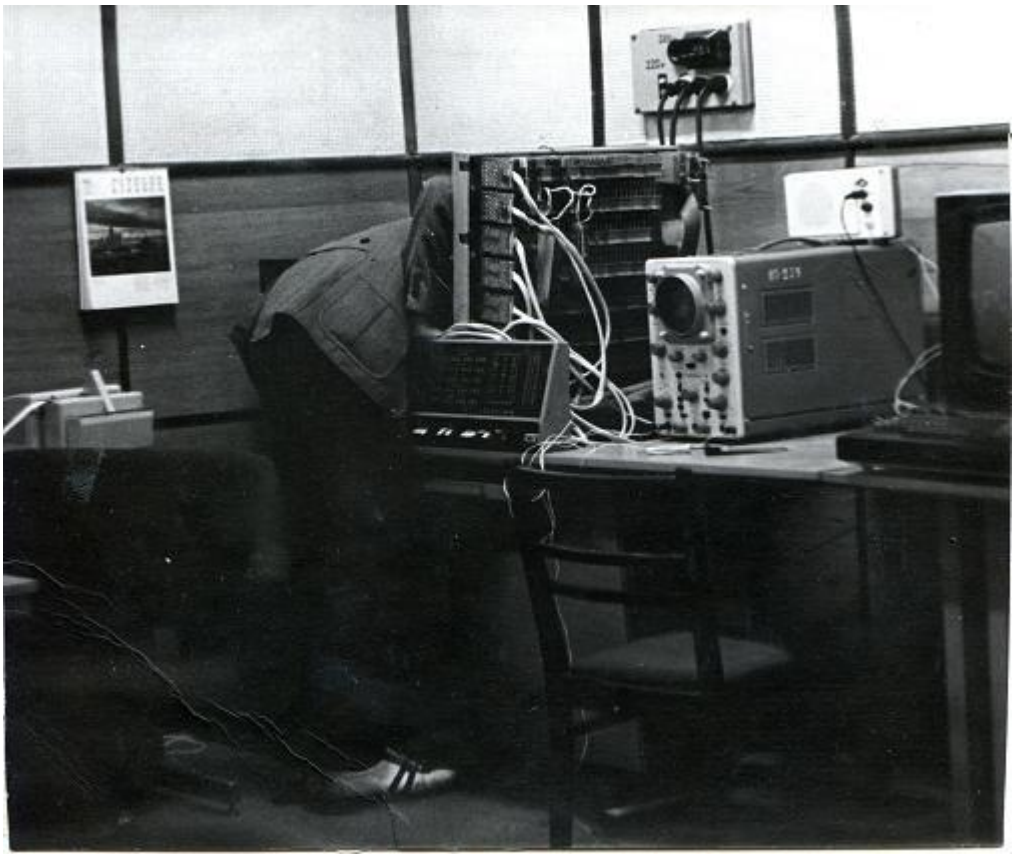
Отладка программы в пошаговом режиме с пульта

Оказывается, первый Windows работал так же

К осени 1978 г. схоластический подход сменился на прагматический. Кафедре было предложено несколько (от 4 до 8) дисплеев - в обмен на обещание сделать терминальный класс. Надо сказать, что «устройства ввода/вывода на основе клавиатуры и электронно-лучевой трубки» были тогда громадным дефицитом, в ходу были элекрифицированные механические пишущие машинки. Не в обиду будет сказано, другая кафедра нашего факультета не стала изобретать велосипед, а пошла по проторенному пути. Взяв за основу микро-компьютер «Электроника-60» с дисковой ОС (кстати сказать, тоже дефицит), она установила на многопользовательский интерпретатор языка Бейсик (мульти-Бейсик). У нас такого компьютера не было, поэтому мы, вслед за Лениным, «пошли другим путем».

Уже готовился проект устройства сопряжения дисплеев с Электроникой-100И, когда от нас потребовалось срочно продемонстрировать руководству результаты работ, чтобы получить «добро» на их продолжение. Худо-бедно, дисплеи были подключены к компьютеру, а вот с софтом дело обстояло хуже. Проще говоря, его еще не было. И тогда мы с **Виталием Любаном** (гр.АМ-79), который с этого времени тоже активно включился в работу, да еще привел с собой талантливых одногруппников – **Геннадия Пестунова** и **Надежду Марьясову** – реализовали простое решение. В известных нам программах – редакторе, отладчике, интерпретаторе Фокала – нашли точки ожидания ввода и заменили в них команды на безусловный переход к диспетчеру. Он, в свою очередь, опрашивал контроллер дисплеев и «рассовывал» полученные данные по программам. То же самое проделали и с выводом. Надо сказать, сработало. К назначенному времени за компьютером сидели 4 пользователя, и каждый работал со своей программой. Единственным «милым» недостатком являлось то, что пользователи не могли переключать программы, и при зависании одной из них приходилось перезагружать всю систему.

Как оказалось значительно позднее, по такому же скользкому пути шел Билл Гейтс в первой версии Windows, там тоже фактически не было разделения времени, и программа сама решала, когда отдавать процессор. Но ему еще за это заплатили.



Если нет ошибки в программе, ищи ее в процессоре

Лирическое отступление. Контроллер экранных пультов

Для сопряжения дисплеев и принтера с компьютером был разработан контроллер с параллельной шиной (нужно учитывать уровень тогдашней схемотехники). Дисплей подключались к шине контроллера через адаптеры, которые выставляли и читали данные с шины по командам контроллера. Контролер поддерживал внутренний алгоритм опроса адаптеров на предмет появления данных, передавал данные от компьютера к адаптерам. Взаимодействовал он в режиме программного обмена по прерыванию. Был заложен также режим блочного обмена в режиме прямого доступа к памяти, когда дисплей мог пересылать весь экран непосредственно в память компьютера. Это усложнило контролер, но, как оказалось, не потребовалось на практике.

Слева направо: Евгений Романов, Виталий Любан, Райнер Гутче

По пути к защищенному режиму

Все-таки разделение времени и защиту нужно было сделать «без дураков». Это оказалось просто. Главное, что организация памяти была блочной (один куб памяти – 4К слов, а всего 8 кубов = 32 К слов, смехотворная цифра по нынешним временам), и в каждом блоке система адресации была исключительно внутренней (вообще-то, при косвенном обращении данные брались из другого блока). Большинство программ работало в одном блоке, лишь некоторые – в двух. Первое, что было сделано, скрутили проводок в селектора команд с кодом **6** – сюда входили ввод-вывод, управление регистрами расширения памяти, разрешение и запрещение прерывания, т.е. то, что обычной программе делать запрещено. Добавили в компьютер схемку – в обычном (системном) режиме она пропускала сигнал выборки этих команд, куда ему полагалось, в «защищенном», вызывала прерывания. Таким образом, можно было создавать полноценное ядро.

Ядро создавало для каждого пользователя свою виртуальную машину. В его функции входило:

- распределение памяти (кубами по 4К слов);
- захват разделяемых устройств (принтера, кассетного магнитофона);
- переключение процессов и простая круговая диспетчеризация;
- загрузка в память пользователя резидентных компонент из памяти ядра (редактора, Ассемблера, отладчика, загрузчика, файловой системы кассетника).

По поводу авторства отдельных программ. Файловую систему накопителя на аудио-кассетах написал **Виталий Любан**. Для этого ему потребовалось 256 слов (2 страницы

памяти). В 512 слов (4 страницы) другой умелец – **Райнер Гутче**, студент из тогдашней ГДР, уместил на место обычного отладчика **полный эмулятор** системы команд компьютера (для этого ему пришлось использовать некоторые команды программы в качестве констант и масок). Дело даже дошло до идей командного процессора, но на том и остановилось. Все в целом и в радужных перспективах именовалось **Многотерминальная Учебная Вычислительная Система (МУВС).**

"Бригада наладчиков" МУВС : Евгений Романов,
Александр Кордонский, Аркадий Исаев, Алексей Дружинин

С каждого user'а – по 128 слов

Надо сказать, что для учебного процесса таких излишеств не требовалось, достаточно было одного Фокала (язык наподобие Бейсика). Фокал занимал почти весь куб (4К слов), оставляя 128 слов под загрузчик. Т.е. в сумме оставалось 1024 свободных слова. В них то и «размазали» ядро, которое при пуске создавало 8 копий Фокала, поддерживало защищенный режим разделения времени, перехват и обработку команд ввода-вывода для дисплеев и принтера, а также захват последнего. Вполне пристойно для 1024 слов.

Геннадий Пестунов и Виталий Любан в "инженерской"

Оказывается, это называется bootstrap

К тому времени на кафедре появилась «настоящая» 16-разрядная мини-ЭВМ с дисковой ОС (правда, оперативки и дисковой памяти было крайне мало). Поэтому на период освоения и комплектования оборудования на нее были возложены почетные обязанности **файлового сервера** PDP-8. Для этого оба компьютера соединили каналом параллельной передачи данных, в СМ-4 повесили программу-сервер, которая по имени файла, полученного из PDP-8, открывала соответствующий файл на диске и побайтно его передавала. В нашей операционке появилось новое виртуальное устройство «ввода/вывода на перфоленту», второе после кассетного накопителя. Надо сказать, что файлы хранились как в текстовом формате, так и в формате вторичного загрузчика с перфолент.

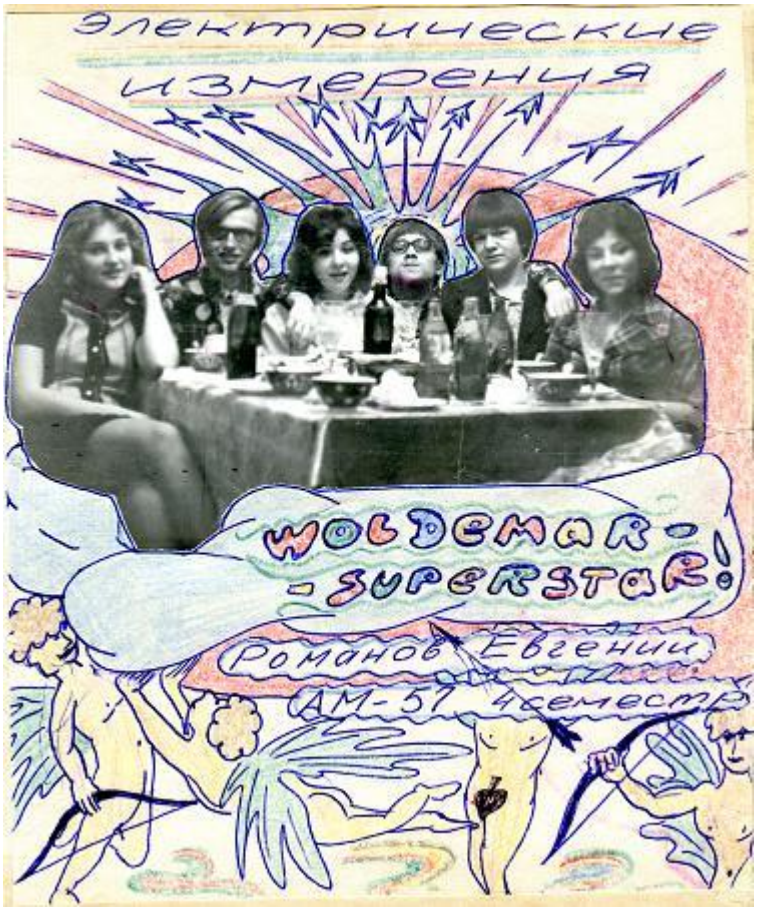
Естественно, что изменилась процедура загрузки и самой операционки. Для ее перезагрузки не нужно было перфоленты, а достаточно было запустить слегка измененный вторичный загрузчик с командами обращения к параллельному каналу, после чего загружался файл операционки и она стартовала. Как оказалось позже, все это называется **bootstap** – процедура раскрутки ОС из загрузочного сектора.

Фотосессия-1979

Из начальников в подчиненные

Однако время брало свое. На СМ-4 была вполне приличная многопользовательская система реального времени – RSX-11M (кстати, ее разработчики потом участвовали в проектировании ядра Windows NT, многие вещи там вполне узнаваемы, например, DPC – «отложенные прерывания»). Единственное, что нам недоставало - у нас не было стандартного мультиплексора терминалов.

Не было, и не надо. Переписали часть драйвера мультиплексора, заменив команды обращения к стандартному «железу» обращением к параллельному каналу, разработали простейший протокольчик, написали простую программу опроса дисплеев и пересылки байтов для PDP-8, и все – наш компьютер из начальника **МУВС** (Многотерминальной Учебной Вычислительной Системы) оказался разжалованным в подчиненные – быть на побегушках у драйвера виртуального мультиплексора.



Конспект лекций по курсу "Электрические измерения" (1976)

Где же вы теперь, друзья однополчане?

Затем, когда появился реальный «железный» мультиплексор, PDP-8 списали на военную кафедру. Чем она там занималась, неизвестно, скорее всего, ее жизнь была не так разнообразна и ее, в конце концов, «разобрали на дрова». Оно, конечно, жаль, что не осталось чего-нибудь для компьютерного музея, либо софта для эмулятора **[4],** но это – как реконструкция Бородинского сражения: красиво, но это уже спектакль, а не жизнь.

1. Методическое руководство по МУВС - методичка НЭТИ
2. http://ru.wikipedia.org/wiki/PDP-8
3. http://comhist.blogs.ru/2007/08/07 - блог Российского виртуального компьютерного музея
4. http://www.vandermark.ch/pdp8/index.php - полный эмулятор PDP-8 на Java с периферией и библиотеками программ образца 1980 года. Ностальгия, однако. К сожалению, нашего софта не сохранилось ни байта.
5. http://www.pcmag.ru/solutions/detail.php?ID=11528 - статья в PCMagazine об истории семейства PDP-8.