

UZI

UZI — это реализация UNIX для процессоров Zilog Z80 («Unix: Z80 Implementation»). Весь код был с нуля написан Дугласом Брауном (Doug Braun) в середине—конце 80х. Благодаря тому, что система не содержала кода, принадлежащего AT&T, она не стала объектом копирайта или лицензионных ограничений со стороны AT&T, и была обнародована как Public Domain.

Как пишет сам Дуглас — «Мне удалось произвести на удивление полную реализацию (Unix v7), несмотря на жалкие 64Кб ОЗУ Z80, отсутствие MMU и аппаратной защиты памяти».

Этот код послужил точкой отсчета для многих других проектов, таких как [UZIX](#), [FUZIX](#), UZI280 etc...

Исходный код был написан в основном на Си, а собирался компилятором The Code Works' Q/C. Множество программ [7th Edition Unix](#) было портировано на UZI совсем — или почти без труда, включая полный Bourne shell, ed, sed, dc, cpr, и т.д.

Архитектура ОС:

Поскольку на компьютерах 8080 семейства нет стандартного механизма менеджмента памяти (MMU), *многопроцессность*¹⁾ достигается методом «total swapping» (полного своппинга). Это подразумевает две вещи: Первое — UZI необходим довольно быстрый жесткий диск. Второе — не имеет смысла запускать другой процесс, пока текущий процесс занят дисковыми операциями. Это упрощает дизайн драйверов блочных устройств, так как от них не требуется использовать прерывания.

Собственно UZI располагается в верхних 32Кб памяти, а работающий в данный момент процесс занимает нижние 32Кб. Так как UZI сейчас едва вмещается в 32Кб, для работы необходимы все 64Кб ОЗУ.

UZI не может обойтись без некоторых дополнительных устройств. Во-первых, должен присутствовать, в том или ином виде, таймер или часы, которые могут обеспечить периодическое прерывание. Также, текущая реализация использует дополнительные часы реального времени (RTC) для получения времени для временных меток файлов, и т.п. Текущий ТТУ-драйвер предполагает наличие клавиатуры, работающей от прерываний, что сегодня²⁾ является нормой на большинстве систем.

В чём UZI отличен от настоящего Unix:

UZI реализует почти весь функционал 7th Edition Unix. Весь файловый ввод-вывод, директории, монтируемые файловые системы, пользовательские и групповые ID, поддерживается конвейер команд и операции с применимыми устройствами. Управление процессами (fork(), execve(), signal(), kill(), pause(), alarm(), and wait()) полностью реализовано. Количество процессов ограничено только размером swap. Как упомянуто ранее, UZI воплощает Unix достаточно хорошо, чтобы исполнять Bourne shell во всей её функциональности. Единственные изменения, внесенные в исходный код shell, были сделаны, чтобы удовлетворить ограничения компилятора Си.

Вот (вероятно, неполный) список ограничений и недостающих функций:

- Отсутствуют системные вызовы, относящиеся к подсистеме отладчика и профайлера.
- Реализован старый 6th edition seek(), вместо lseek().
- Поставляемый ТТУ-драйвер это лишь костяк. Он поддерживает только один порт, и большинство IOCTL не поддерживаются.
- Номера l-нодов всего лишь 16-битные, так что файловые системы — 32Мб или меньше.
- Даты файлов не в стандартном формате. Вместо этого они больше похожи на те, что используются в MS-DOS.
- Реализован 4.2BSD execve(). Дополнительные варианты exec() поддерживаются библиотекой.
- Формат таблицы ключей драйверов устройств не такой, как в 7th Edition.
- Необходимые семафоры и механизмы блокировок для реализации реентерабельного дискового ввода-вывода отсутствуют. Это затруднит реализацию основанного на прерываниях дискового ввода-вывода.

Примечание 1

Не используем термин *многозадачность*, поскольку здесь это не вполне корректно. Хотя, с пользовательской точки зрения, эффект достигается почти тот же. Примечание 2

Данная статья является, большей частью, почти дословным переводом оригинального [текста-анонса](#), сделанного Дугом Брауном 8 ноября 1988 в USENET.

Ссылки



<https://www.dougbraun.com/uzi.html>

<https://groups.google.com/g/comp.os.cpm/c/2YGLU25TboA>

¹⁾

Смотри [Примечание 1](#)

²⁾

Смотри [Примечание 2](#)

<http://sysadminmosaic.ru/uzi/uzi>

2022-09-01 13:51

