

## ЛИТЕРАТУРА

1. Агibalов Г. П., Липский В. Б., Панкратова И. А. О криптографическом расширении и его реализации для Русского языка программирования // Прикладная дискретная математика. 2013. №3. С. 93–105.
2. Netwide Assembler. <http://www.nasm.us/>
3. TIS Committee. Tool Interface Standard (TIS) Executable and Linking Format (ELF) Specification Version 1.2. 1995.
4. The GNU linker ld (GNU Binutils) version 2.25. <https://sourceware.org/binutils/docs-2.25/ld/index.html>

УДК 004.451.9

DOI 10.17223/2226308X/8/52

## РАЗРАБОТКА ОПЕРАЦИОННОЙ СИСТЕМЫ НА ЯЗЫКЕ ЛЯПАС

Д. А. Стефанцов, П. А. Томских

Сообщается о результатах исследований по созданию ОС ЛЯПАС, предназначенной для разработки и запуска программ на ЛЯПАСе. Реализованы следующие функции её ядра: вывод сообщений на экран, инициализация контроллеров устройств, обработка прерываний, взаимодействие с таймером и клавиатурой, многозадачность. Многозадачность демонстрируется несколькими параллельно работающими процессами, которые можно приостанавливать и возобновлять нажатием определённой клавиши. Планируется реализация файловой подсистемы и подсистемы виртуальной памяти.

**Ключевые слова:** *ЛЯПАС, операционная система.*

ОС ЛЯПАС предназначена для разработки и запуска программ, написанных на языке ЛЯПАС [1]. Предполагается специализация данной ОС для выполнения требовательных к вычислительным ресурсам криптографических алгоритмов. Планируется работа ОС ЛЯПАС на процессорах архитектуры x86-64 с возможностью использовать аппаратные средства для ускорения работы алгоритмов, например выделить одно или несколько ядер процессора полностью под один процесс или использовать графические процессоры для параллельных вычислений. В настоящий момент ОС ЛЯПАС компилируется для 32-битных процессоров архитектуры x86.

В разработке этой ОС можно выделить следующие этапы:

- 1) Создание транслятора T1 с ЛЯПАСа в язык ассемблера, работающего под управлением GNU/Linux. ЛЯПАС, с которого производится трансляция, обладает дополнительными возможностями по сравнению с ЛЯПАС-Т [1]: доступ к любой ячейке памяти с помощью специального комплекса, возможность использовать адреса процедур в качестве операндов [2].
- 2) Создание базовых компонент ОС ЛЯПАС — загрузчика и ядра — с помощью T1. После компиляции в машинные коды и расположения этих компонент на диске ОС ЛЯПАС может выполнять некоторые действия, например вывод на экран данных о своей работе.
- 3) Определение формата исполняемого файла для ОС ЛЯПАС. Возможно, таким форматом станет ELF [3].
- 4) В ОС ЛЯПАС добавляется возможность запуска программ. Эти программы пишутся в GNU/Linux на языке ЛЯПАС и транслируются в машинный код, представленный в формате исполняемого файла ОС ЛЯПАС, с помощью транслятора T2. Транслятор T2 получается изменением T1.

- 5) Под управлением GNU/Linux пишется транслятор ТЗ, который транслирует программы на ЛЯПАСе в машинный код в формате исполняемого файла для ОС ЛЯПАС. Сам ТЗ пишется на ЛЯПАСе и транслируется с помощью Т2. После этого шага становится возможным создание программ для ОС ЛЯПАС в самой ОС ЛЯПАС.
- 6) Исходный текст ОС ЛЯПАС и транслятор ТЗ изменяются так, чтобы ТЗ мог скомпилировать ОС ЛЯПАС. После этого шага ОС ЛЯПАС можно разрабатывать в самой ОС ЛЯПАС.

На данном этапе разрабатываются загрузчик и ядро ОС ЛЯПАС. В функции загрузчика входят только очистка экрана и загрузка ядра с диска.

Ядро разделено на два модуля. В первом модуле производится переход в защищённый режим и вызов второго модуля. В этом модуле определена также глобальная таблица дескрипторов (GDT) и выделяется память под таблицу дескрипторов прерываний (IDT) [4].

Основные функции ядра находятся во втором модуле. К таким функциям относятся: вывод строк на экран (функция `print(F1/)`, комплекс F1 — выводимая строка); инициализация контроллеров (функция `init_controllers(/)`); обработка прерываний (функция `timer_interrupt(/)` обрабатывает прерывания от таймера, функция `keyboard_interrupt(/)` — от клавиатуры); заполнение IDT (функция `construct_idt(/)`); переключение процессов (при нажатии клавиши 1, 2 или 3 происходит остановка или возобновление соответствующего процесса, выполняется это функцией `handle_scancode(/)`, которая вызывается функцией `keyboard_interrupt(/)`); многозадачность (функция `timer_interrupt(/)` вызывает планировщик — функцию `scheduler(/)`, который переходит на выполнение следующего процесса). В этом же модуле находится код процессов, демонстрирующих многозадачность (функции `proc_1(/)`, `proc_2(/)`, `proc_3(/)` — бесконечный цикл с выводом названия соответствующей функции, и `proc_0(/)` — бесконечный цикл без действий). Подготовка к переходу к первому процессу осуществляется функцией `kernel(/)`.

На данный момент ОС ЛЯПАС после загрузки переходит к демонстрации многозадачности — поочерёдному выполнению трёх процессов (функции `proc_1(/)`, `proc_2(/)` и `proc_3(/)`). Эти процессы можно приостанавливать и возобновлять нажатием клавиш 1, 2 или 3 соответственно. Если эти три процесса приостановлены, то выполняется процесс `proc_0(/)`.

В ближайшее время планируется добавить некоторые системные вызовы, настроить страничную адресацию, реализовать работу с файлами.

## ЛИТЕРАТУРА

1. Агibalов Г. П., Липский В. Б., Панкратова И. А. О криптографическом расширении и его реализации для Русского языка программирования // Прикладная дискретная математика. 2013. № 3. С. 93–105.
2. Гречнев С. Ю., Стефанцов Д. А. Модификация ЛЯПАС для разработки ОС // Прикладная дискретная математика. Приложение. 2015. № 8. С. 129–131.
3. TIS Committee. Tool Interface Standard (TIS) Executable and Linking Format (ELF) Specification Version 1.2. 1995.
4. Intel 64 and IA-32 Architectures Software Developer Manuals <http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html>