

Разработка программного комплекса для автоматизированной установки операционной системы реального времени «АxiOMA RTOS»¹

Р.Л. Пушков,
ст. преп., pushkov@ncsystems.ru,
С.В. Рыбников,
зав.лаб., rybnikov@ncsystems.ru,
А.Н. Нукич,
магистр 1-го года обуч., n_nikich@me.com,
МГТУ «СТАНКИН», г. Москва

Проанализированы существующие способы установки операционных систем на базе ядра Linux. Выявлены основные и дополнительные шаги установки для использования в инсталляторе с учетом специфики ОС реального времени «АxiOMA RTOS». Создана блок-схема алгоритма разработки инсталлятора. Разработаны специальные скрипты для программной реализации, позволяющие разворачивать систему на чистом дисковом пространстве. Собран экспериментальный стенд для аппаратной реализации, демонстрирующий практическое применение инсталлятора на конкретном оборудовании.

Analyzed the existing methods of installation of operating systems based on the Linux kernel. Basic and advanced installation steps are identified for use in the installer in view of specificity of real-time operating system «АxiOMA RTOS». Creation the model of the algorithm development of the installer. For a software implementation of the installer the specifically scripts were written to allow to deploy the system on a clean disk space, and for hardware implementation assembled experimental stand which demonstrates the practical application on the specific equipment.

Управление автоматизированными комплексами, где от времени выполнения задач напрямую зависит требуемая точность и качество обработки, осуществляется с помощью операционных систем реального времени (ОСРВ). Основными задачами ОСРВ являются:

1. Обеспечить гарантированное время реакции на внешние события;
2. Обеспечить жёсткую подсистему диспетчеризации процессов;
3. Обеспечить предсказуемость поведения при любых сценариях нагрузки [1].

ОС «АxiOMA RTOS» для системы ЧПУ «АксиОМА Контрол» относится к системам жёсткого реального времени. Она построена на базе ядра ОС GNU/Linux с использованием патча RT-PREEMPT, который позволяет из обычной операционной системы Linux сделать операционную систему жесткого реального времени [2].

Патч RT-PREEMPT вносит ряд изменений в базовое ядро Linux. Во-первых, он преобразует все ISR (Interrupt Service Routines) в потоки, называемые «Threads Interrupt Service Routines». Во-вторых, заменяет принцип синхронизации spinlock на семафорный механизм (mutex), который поддерживает наследование приоритетов и вытеснение. В третьих, он добавляет поддержку режима High Resolution Timer (HRT), что позволяет таймерам работать с разрешением 1 мкс. Кроме того, он обеспечивает защиту от возникновения неограниченной инверсии приоритетов [6].

Структурная схема системы управления представлена на рис. 1. Пульт оператора связывается со шкафом управления по двум интерфейсам - TCP/IP, который обеспечивает взаимодействие машины реального времени под управлением Linux RT и терминального компьютера под управлением Windows XP; RS-485, обеспечивающего работу станочной панели в режиме жесткого реального времени. Функциональная клавиатура и промышленная клавиатура работают в машинном времени и подключены к терминальному компьютеру (по интерфейсам RS-485 и USB соответственно).

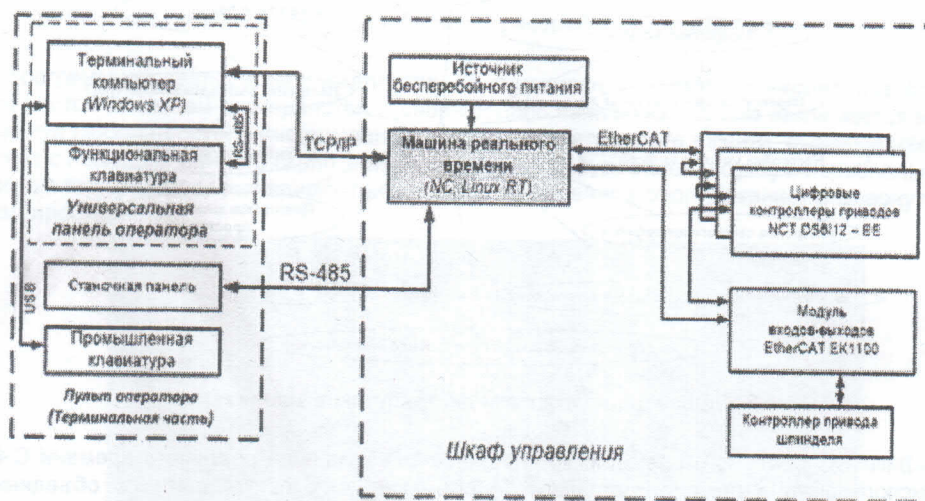


Рис.1 Структурная схема системы ЧПУ «АксиОМА Контрол»

¹ Работа выполнена при финансовой поддержке Министерства образования и науки РФ в рамках государственного задания в сфере научной деятельности.

Терминальный компьютер, функциональная клавиатура, станочная панель и промышленная клавиатура вместе составляют пульт оператора.

Шкаф управления состоит из следующих компонентов: машина реального времени (выполняет функцию ядра системы ЧПУ, работает в режиме жесткого реального времени и решает геометрическую, логическую, технологическую задачи управления), источник бесперебойного питания, приводная часть и электроавтоматика [3,4].

При создании собственного программного обеспечения перед любым разработчиком возникает задача переноса своей программы, приложения или системы на компьютер конечного пользователя, избавив его при этом от каких-либо действий, которые требуют высокого уровня квалификации по её установке. После создания собственного продукта, необходимо решить вопрос, каким образом он будет устанавливаться конечным пользователем.

Задача автоматизации установки операционной системы реального времени сводится к минимизации шагов пользователя по ее инсталляции на компьютер.

В качестве средства автоматизации установки выступает специально разработанная программа, которая называется инсталлятором, целью которой является упрощение процесса развертывания операционной системы реального времени с минимальным интерфейсом на чистом дисковом пространстве.

Инсталлятор должен решать следующие задачи:

1. Создание программы для установки приложения, выполняющую инсталляцию;
2. Создание программы для удаления приложения из системы — выполняющую процедуру деинсталляции;
3. Сжатие и упаковка файлов, входящих в дистрибутив, распаковка и размещение файлов в необходимые для установки места.

Проанализировав решения различных компаний можно прийти к выводу, что, в данной ситуации, они не подходят для инсталляции операционной системы реального времени, так как все они подразумевают наличие уже заранее установленной операционной системы или работу с прикладным ПО. В рамках поставленной задачи пользователь может установить систему на компьютер без какой-либо заранее предустановленной ОС. Для создания такого инсталлятора необходимо создание собственной минимальной системы и написание специальных скриптов для его запуска. Под созданием собственной минимальной системы подразумевается выбор любого минимального дистрибутива Linux.

В данном случае был выбран Linux From Scratch (дословно переводится как "Линукс с нуля"), который позволяет собрать и сконфигурировать свою Linux-систему со своими параметрами.

Выбор Linux From Scratch обусловлен прежде всего тем, что данный дистрибутив является одним из наиболее минимальных по объему занимаемой памяти и позволяет осуществлять точную настройку системы [5].

При создании инсталлятора были проанализированы различные ОС на базе ядра Linux (Ubuntu, Fedora, Gentoo, Debian).

В ходе анализа перечисленных систем были выявлены следующие общие шаги установки (рис. 2):

1. Выбор языка для операционной системы;
2. Выбор часового пояса;
3. Настройка раскладки клавиатуры;
4. Выбор местонахождения;
5. Подготовка дискового пространства;
6. Создание разделов диска (проверка наличия свободного места на диске, освобождение места под разделы, создание разделов, разделов подкачки, выбор файловой системы);
7. Выбор устанавливаемого программного обеспечения;
8. Загрузка дополнительных компонентов (если требуется);
9. Настройка сетевого интерфейса компьютера;
10. Настройка учетных записей (регистрация нового пользователя);
11. Перенос настроек из других операционных систем (если они имеются);
12. Завершение установки.

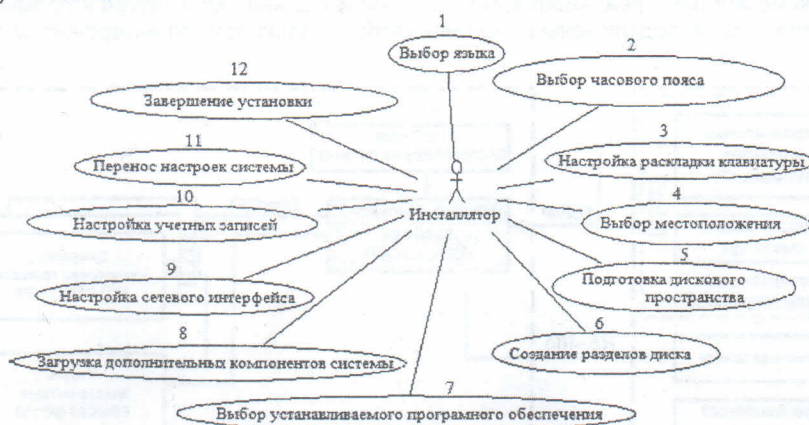


Рис.2 Общие шаги установки дистрибутивов Linux-систем

С учетом специфики ОС реального времени, установленной на машине реального времени СЧПУ «АксиОМА Контрол», решено использовать в инсталляторе шаги 2,4, объединив их в шаг "а", шаги 5,6, объединив их в шаг "б", шаги 1,3,7,9,10 и 12 применяются без изменений (рис. 3) шаги 8,11 исключаются.

Основные шаги установки необходимы для установки ОС и базового набора ПО для использования СЧПУ и эксплуатации станка:

- а) Установка системного времени;
- б) Подготовка дискового пространства и разделов диска (проверка наличия свободного места на диске, освобождение места под разделы, создание разделов, разделов подкачки, выбор файловой системы);

- в) Настройка сетевого интерфейса компьютера;
- г) Настройка учетных записей (регистрация нового пользователя);
- д) Завершение установки.

Дополнительные шаги установки необходимы для установки дополнительного ПО, разработки и тестирования ядра СЧПУ:

- е) Выбор языка для операционной системы;
- ж) Настройка раскладки клавиатуры;
- з) Выбор устанавливаемого программного обеспечения.

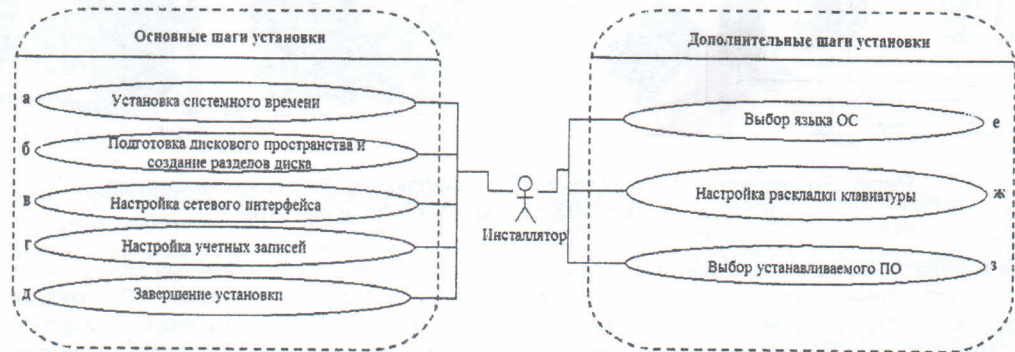


Рис.3 Разработанные основные и дополнительные шаги установки с учетом специфики ОС реального времени «AxiOMA RTOS»

Предложена блок-схема алгоритма разработки инсталлятора (рис. 4):

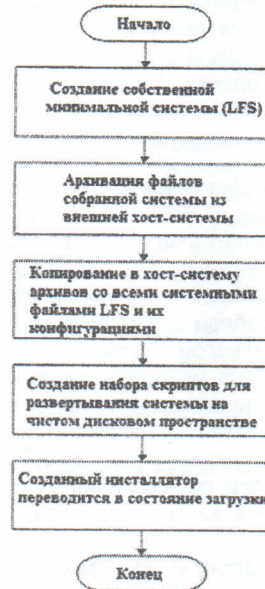


Рис.4 Блок-схема алгоритма разработки инсталлятора

В качестве программной реализации инсталлятора был написан специальный скрипт, позволяющий разворачивать систему на чистом дисковом пространстве. Основная цель написания скрипта заключается в следующем: пользователь, используя любой носитель информации (флэш-накопитель, карта памяти, съёмный жёсткий диск или компакт - диск) может с помощью данного скрипта на любом типе оборудования (ноутбук, персональный компьютер, одноплатный компьютер, учебное, экспериментальное и промышленное оборудование) развернуть ядро системы на чистом дисковом пространстве.

```

cp -r libKernelTools.so /usr/lib
cp libKernelTools.so /usr/lib/libKernelTools.so
nkernel1: Выход из каталога "/sources/Kernel/KernelTools"
nkernel1: Выход в каталог "/sources/Kernel/Kc"
cp -r libKc.so /usr/lib
cp libKc.so /usr/lib/libKc.so
nkernel1: Выход из каталога "/sources/Kernel/Kc"
nkernel1: Выход в каталог "/sources/Kernel/Kernel"
cp -r libKc /usr/lib
cp libKc /usr/lib/libKc
nkernel1: Выход из каталога "/sources/Kernel/Kernel"
nkernel1: Выход в каталог "/sources/Kernel/Kernel"
nkernel1: Выход в каталог "/sources/Kernel/Kernel"
Kernel install complete
Kernel
Kernel 2015_01_21
Kernel 2015_01_26_21
Kernel 2015_02_03
Kernel 2016_05_19
Sources Kernel oas deleted
zboot/sdhd on /media/floppy type vfat (rw,relatime,fsck=0022,dev=0022,codepage=cp437,uid=0,gid=0,inode64,errors=remount-ro)
fsckfs: 2015_01_21
Kernel 2016_02_00_21 Kernel 2015_01_26_21 Kernel 2016_05_19 nkernel 0.9.7
Kernel 2015_01_21 Kernel 2015_02_03 linux-3.6.11 test.sh
fsck 4.28
  
```

Рис.5. Выполнение скрипта и установка ядра системы «AxiOMA RTOS»

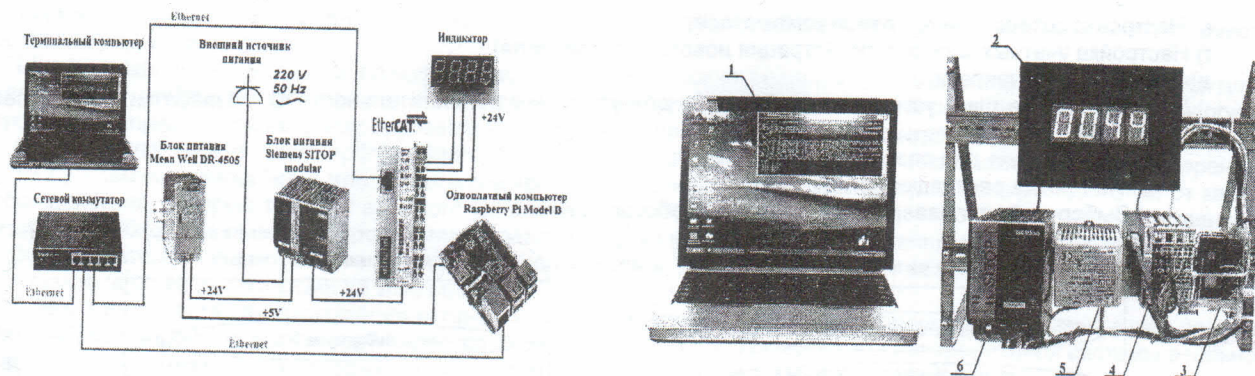


Рис.6. Экспериментальный стенд, представляющий работу инсталлятора и структурная схема соединения его элементов

В качестве аппаратной реализации инсталлятора был разработан экспериментальный стенд на котором представлены: 3 - Одноплатный компьютер Raspberry, куда устанавливается ядро системы ЧПУ, который эмулирует работу машины реального времени. 4 - Модуль входа-выхода Beckhoff EK1100, связанный с Raspberry по высокоскоростному протоколу EtherCAT. Модуль входа-выхода эмулирует работу ПЛК. Модуль входа-выхода служит только для вывода сигнала, так как все вычисления выполняются на Raspberry. Информация с модуля входа-выхода выводится на индикатор – 2. Блок питания Siemens SITOP modular 24В - 6 питает модуль входа-выхода. Блок питания Mean Well DR - 4505 5В - 5 питает одноплатный компьютер Raspberry. Терминальный компьютер - 1 связан с одноплатным компьютером Raspberry по кабелю Ethernet. Терминальный компьютер служит для запуска скриптов и визуализации результата. Для соединения с одноплатным компьютером Raspberry по сетевому протоколу SSH используем PuTTY клиент. PuTTY - свободно распространяемый клиент для различных протоколов удалённого доступа, включая SSH, Telnet, rlogin. Также имеется возможность работы через последовательный порт. PuTTY позволяет подключиться и управлять удаленным узлом (например, сервером).

Заключение

В ходе выполнения данной работы для разработки архитектуры инсталлятора были проанализированы существующие способы установки операционных систем на базе ядра Linux, выявлены общие шаги установки дистрибутивов Linux-систем, определены основные и дополнительные шаги установки разрабатываемого инсталлятора с учётом специфики реального времени, создана блок-схема алгоритма разработки инсталлятора. Для программной реализации инсталлятора был написан специальный скрипт, позволяющий разворачивать систему на чистом дисковом пространстве, а для аппаратной реализации собран экспериментальный стенд, демонстрирующий практическое применение на конкретном оборудовании. Произведено тестирование работы инсталлятора на персональном компьютере и одноплатном компьютере. Разработана методика построения инсталлятора для операционных систем реального времени на базе ядра Linux с применением скрипта, учитывающего специфику реального времени.

Предложенный подход автоматизированной установки ядра операционной системы реального времени «АxiOMA RTOS» для установки, создания и построения систем управления технологического оборудования, отличается применением специально разработанного скрипта, осуществляющего разворачивание системы на чистом дисковом пространстве, и позволяет существенно сократить время разработки системы управления, а также время на пусконаладочные работы.

Практическая значимость данной работы заключается в разработке программного комплекса на базе скрипта, позволяющего устанавливать ядро операционных систем реального времени для систем управления на чистом дисковом пространстве, а также в разработке экспериментального стенда, состоящего из одноплатного компьютера Raspberry, куда устанавливается ядро системы ЧПУ, модуля входа-выхода, индикатора, блока питания на 24В, блока питания на 5В и терминального компьютера.

Приведенное решение позволяет существенно снизить квалификационные требования к пользователю, осуществляющему установку системы, а также время, затраченное на построение системы. В качестве потребителей разработанного решения могут выступать станкостроители, разработчики систем ЧПУ, работающие совместно со станкостроителями по установке и адаптации системы ЧПУ к станочному оборудованию, а также специалисты, осуществляющие техническую поддержку оборудования удалённо с использованием SSH соединения.

Литература

1. Сосонкин В.Л., Мартинов Г.М. Программирование систем числового программного управления: учеб. пособие. - М. Логос, 2008. - 344 с. + компакт-диск. ISBN 978-5-98704-296-8.
2. Сосонкин В.Л., Мартинов Г.М. Системы числового программного управления: Учеб. пособие. - М. Логос, 2005. - 296 с. ISBN 5-98704-012-4.
3. Мартинов Г.М., Мартинова Л.И., Козак Н.В., Нежметдинов Р.А., Пушков Р.Л. Принципы построения распределенной системы ЧПУ технологическими машинами с использованием открытой модульной архитектуры // Справочник. Инженерный журнал. 2011, №12.
4. Мартинов Г.М., Никишечкин П.А., Григорьев А.С., Червонова Н.Ю. Организация взаимодействия основных компонентов в системе ЧПУ АксиОМА Контроль для интеграции в нее новых технологий и решений // Автоматизация в промышленности. 2015. №5. с.10-15.
5. Инструкция по сборке минимальной системы <http://wiki.linuxfromscratch.org/lfs>.
6. Инструкция по установке патча реального времени https://rt.wiki.kernel.org/index.php/RT_PREEMPT_HOWTO.