

Теоретические аспекты разработки программного комплекса для автоматизированной установки операционной системы реального времени «AxiOMA RTOS»

Theoretical aspects of development software system to automate the installation real time operating system «AxiOMA RTOS»

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 AxiOMA RTOS. Создана блок-схема алгоритма разработки инсталлятора.

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. Creation the model of the algorithm development of the installer.

Ключевые слова: операционная система, реальное время, патч реального времени, дистрибутивы, шаги установки, Linux-система, инсталлятор, алгоритм разработки инсталлятора.

Keywords: real-time operating systems, real time patch RT-PREEMPT, distributions based on the Linux kernel, steps for installing Linux-systems, installer, installer algorithm development.

Управление автоматизированными комплексами, в которых требуемые точность и качество обработки напрямую зависят от времени выполнения задач, осуществляется с помощью операционных систем реального времени (ОСРВ).

Основные задачи ОСРВ заключаются в обеспечении:

- 1) гарантированного времени реакции на внешние события;
- 2) жесткой подсистемы диспетчеризации процессов;
- 3) предсказуемости поведения при любых сценариях нагрузки.

Операционная система (ОС) AxiOMA RTOS для системы ЧПУ АксиОМА Контрол относится к системам жесткого реального времени. Она построена на базе ядра ОС GNU/Linux с использованием патча RT-PREEMPT, который позволяет из обычной ОС Linux сделать операционную систему жесткого реального времени [1, 2].

Патч RT-PREEMPT вносит ряд изменений в базовое ядро Linux. Во-первых, он преобразует все ISR

(Interrupt Service Routines) в потоки 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 неограниченной инверсии приоритетов [3].

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

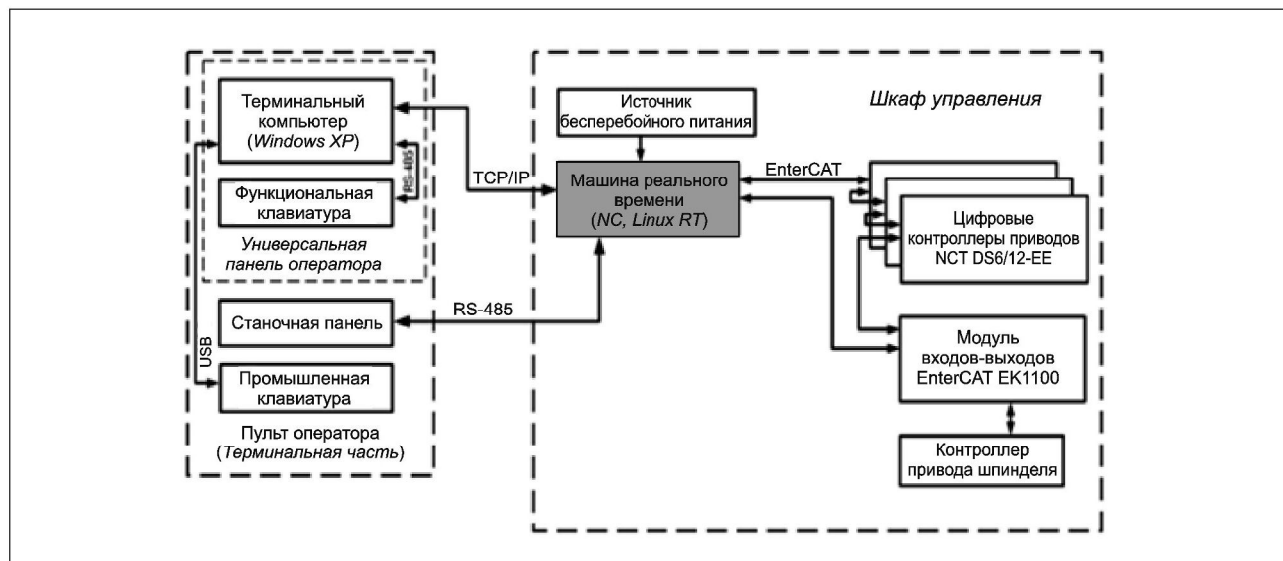


Рис. 1. Структурная схема системы ЧПУ АксиОМА Контрол

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

Шкаф управления состоит из машины реального времени (функция ядра системы ЧПУ; работает в режиме жесткого реального времени; решает геометрические, логические, технологические задачи управления); источника бесперебойного питания; приводная часть; электроавтоматики [4, 5].

При создании собственного программного обеспечения перед разработчиком стоит задача переноса собственной программы, приложения или системы на компьютер конечного пользователя, при этом исключив действия, требующие высокой квалификации по установке. После создания собственного продукта решается вопрос — как его будет устанавливать пользователь.

Автоматизация установки ОСВР сводится к минимизации шагов по ее установке на компьютер. В качестве средства автоматизации установки используют специально разработанную программу, так называемый инсталлятор, цель которого — упростить процесс развертывания ОСВР 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.

Анализ решений разных разработчиков показал, что в данном случае они не подходят для инсталляции ОСВР, так как подразумевают наличие уже заранее установленной ОС 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 необходима собственная минимальная система и написание специальных скриптов для пуска. Под созданием собственной минимальной системы подразумевается выбор любого минимального дистрибутива Linux.

В данном случае выбран Linux From Scratch (Линукс с нуля), который позволяет собрать и сконфигурировать собственную Linux-систему со своими параметрами.

Выбор Linux From Scratch обусловлен прежде всего тем, что данный дистрибутив является наиболее минимальным по объему памяти и позволяет осуществлять точную настройку системы [6].

Были исследованы разные ОС на базе ядра Linux: Ubuntu, Fedora, Gentoo и Debian. Их анализ позволил выявить следующие общие шаги установки:

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.

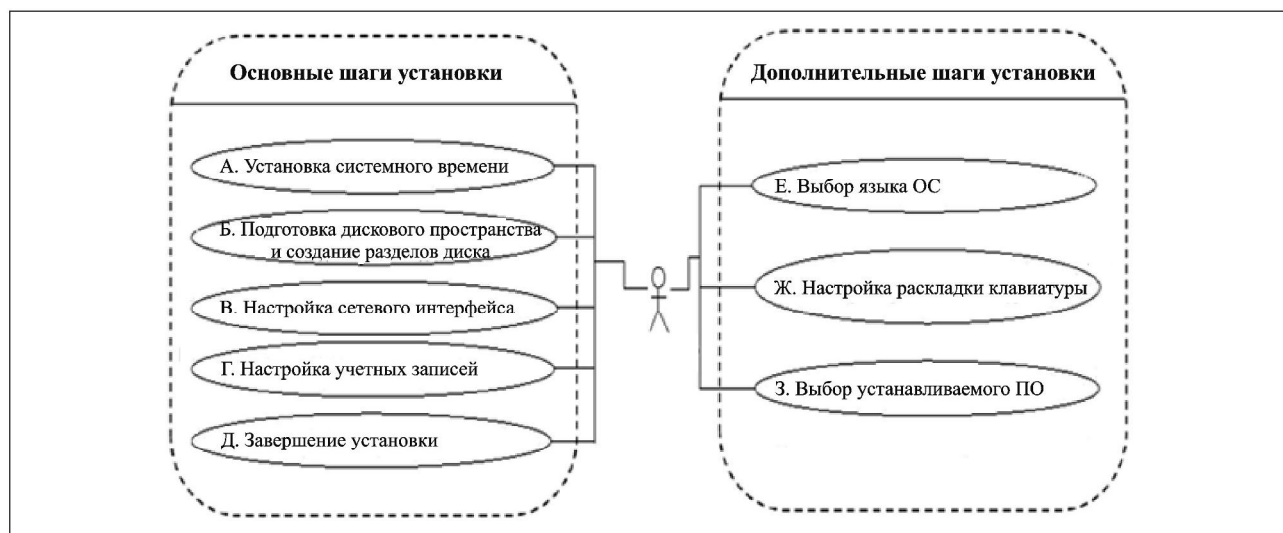


Рис. 2. Основные и дополнительные шаги установки с учетом специфики OSCPv AxiOMA RTOS

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.

С учетом специфики OSCPv, установленной на машине реального времени СЧПУ АксиОМА Контроль, в инсталляторе было решено использовать: шаги 2 и 4, объединив их в шаг А; шаги 5 и 6, объединив их в шаг Б; шаги 1, 3, 7, 9, 10 и 12 без изменений (рис. 2). Шаги 8 и 11 были исключены.

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

Подготовка дискового пространства и разделов диска включает в себя проверку наличия свободного места на диске, освобождение места под разделы, создание разделов, разделов подкачки, выбор файловой системы.

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

Таким образом, предлагается следующий алгоритм разработки инсталлятора:

1. На базе имеющейся Linux-системы, например хост-системы Ubuntu, создание собственной минимальной системы(LFS).

2. Архивирование файлов системы LFSиз внешней хост-системы, например Ubuntu.

3. Копирование в хост-систему архивов со всеми системными файлами LFS и их конфигурациями.

4. Создание набора скриптов, который позволяет развернуть данную систему на чистом дисковом пространстве.

5. Выполнение созданного инсталлятора загрузочным.

Библиографический список

1. Сосонкин В.Л., Мартинов Г.М. Программирование систем числового программного управления: Уч. пособ. — М.: Логос, 2008. — 344 с.

2. Сосонкин В.Л., Мартинов Г.М. Системы числового программного управления: учеб. пособ. — М.: Логос, 2005. — 296 с

3. Инструкция по установке патча реального времени. — Режим доступа: https://rt.wiki.kernel.org/index.php/RT_PREEMPT_HOWTO.

4. Мартинов Г.М., Мартинова Л.И. Формирование базовой вычислительной платформы ЧПУ для построения специализированных систем управления // Вестник МГТУ «Станкин». 2014. № 1 (24). С. 92—97.

5. Организация взаимодействия основных компонентов в системе ЧПУ АксиОМА Контроль для интеграции в нее новых технологий и решений /Г.М. Мартинов, П.А. Никищечкин, А.С. Григорьев, Н.Ю. Червонова // Автоматизация в промышленности. 2015. № 5. С. 10—15.

6. Инструкция по сборке минимальной системы. — Режим доступа: <http://wiki.linuxfromscratch.org/lfs>.

7. Мартинов Г.М., Мартинова Л.И., Пушков Р.Л. Автоматизация технологических процессов в машиностроении: учеб. пособ. — М.: МГТУ «Станкин», 2011. — 200 с.

8. Мартинов Г.М., Мартинова Л.И., Пушков Р.Л. Системы числового программного управления для автоматизации технологических процессов машиностроения

тельного комплекса России: учеб. пособ. — М.: МГТУ «Станкин», 2011. — 169 с.

9. Григорьев С.Н. Принципы создания многофункциональной системы числового программного управле-

ния технологическим оборудованием на базе общего ядра с открытой модульной архитектурой // Приборы и системы. Управление, контроль, диагностика. 2011. № 5. С. 1—11.

Никишечкин Анатолий Петрович — канд. техн. наук, доцент кафедры компьютерных систем управления МГТУ «СТАНКИН».

Тел.: 8(915) 325-51-32. E-mail: anatoij-petrovich@yandex.ru

Пушков Роман Львович — ст. преподаватель кафедры компьютерных систем управления МГТУ «СТАНКИН».

Тел.: 8(499) 972-94-40. E-mail: pushkov@ncsystems.ru

Никич Анатолий Николаевич — студент факультета «Информационные технологии и системы управления» кафедры компьютерных систем управления МГТУ «СТАНКИН».

Тел.: 8(916) 064-00-76. E-mail: n_nikich@me.com

Nikishechkin Anatoliy Petrovich — Candidate of Engineering Sciences, Associate professor of the sub-department «Computer control systems» of MSTU «STANKIN».

Tel.: +7(915) 325-51-32. E-mail: anatoij-petrovich@yandex.ru

Pushkov Roman Lvovich — senior lecturer of the sub-department «Computer control systems» of MSTU «STANKIN».

Tel.: +7(499) 972-94-40. E-mail: pushkov@ncsystems.ru

Nikich Anatoliy Nikolaevich — fourth-year student of the Faculty «Information technology and control systems», sub-department «Computer control systems» of MSTU «STANKIN».

Tel.: +7(916) 064-00-76. E-mail: n_nikich@me.com
