

Московский физико-технический институт  
(государственный университет)  
Факультет радиотехники и вычислительной техники  
Кафедра информатики и вычислительной техники

Выпускная квалификационная работа бакалавра

# **АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ БИБЛИОТЕКИ ЗАЩИЩЕННОГО РЕЖИМА ИСПОЛНЕНИЯ ПРОГРАММ АРХИТЕКТУРЫ “ЭЛЬБРУС”**

Студент: Данилин М.Н.

Научный руководитель: Шигунов Ф.Г.

# Постановка задачи

Автоматизация системы тестирования для прохождения испытаний, предназначенных для проверки работоспособности библиотеки защищенного режима исполнения программ на разных версиях ядра ОС «Эльбрус-Д».

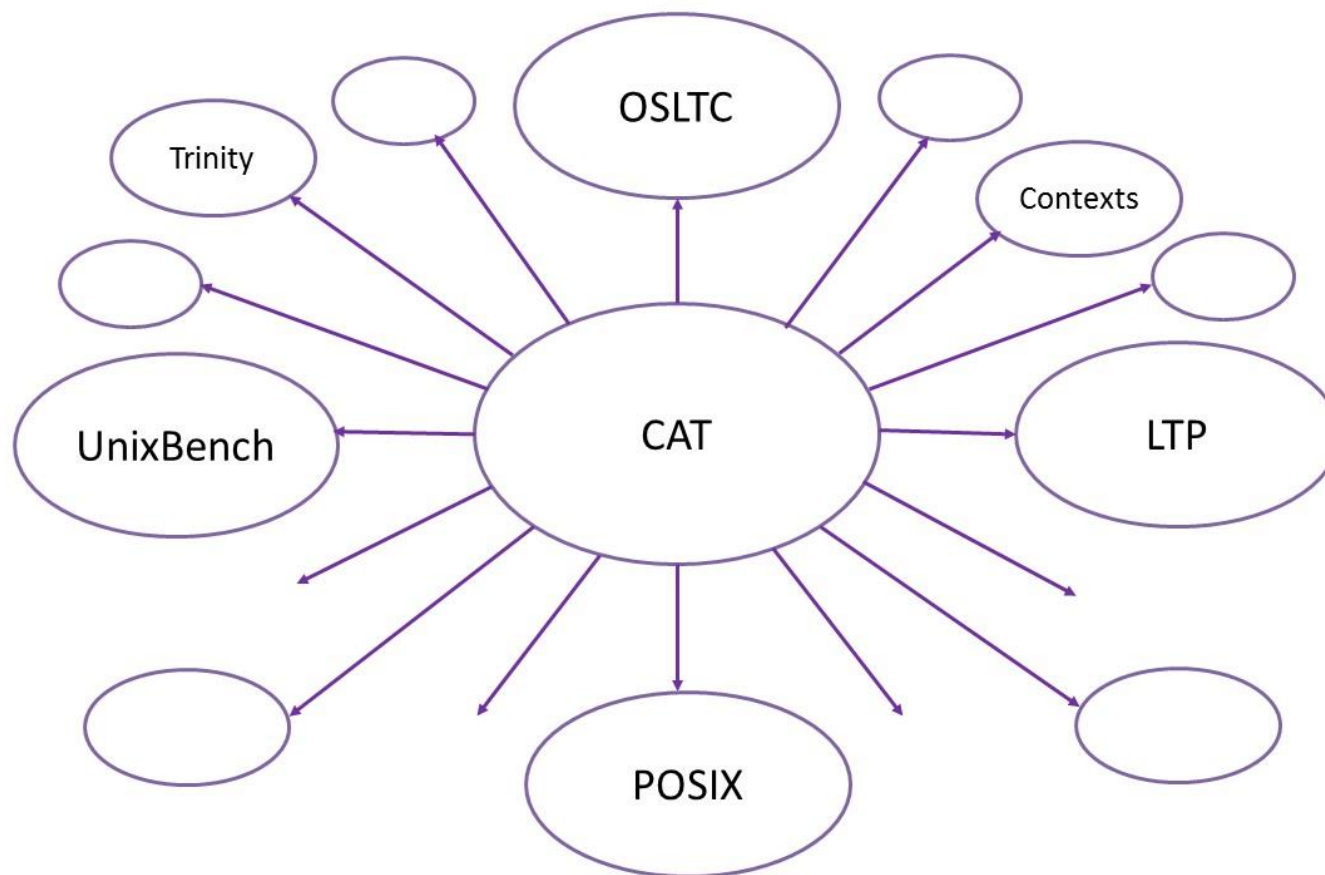
## Требования к разработке

- ▶ Создать систему автоматизации тестирования(CAT) ядер ОС «Эльбрус-Д»
- ▶ Поддержать возможность включения в сценарии тестирования сторонних групп тестов
- ▶ Проведение тестирования как в обычном, так и в защищенном режимах исполнения программ
- ▶ Реализовать возможность проверки разных конфигураций и разных версий ядра на базе архитектуры Эльбрус

# Метод решения задачи

- ❑ Использование утилиты Autotest для создания системы автоматизированного тестирования
- ❑ Добавление в Autotest функционала по:
  1. поиску и установке новой версии ядра ОС
  2. запуску системы тестирования ОС
  3. обработке и сохранению результатов в архив

# Состав системы тестирования ОС «Эльбрус-Д»



# Тестирование библиотеки защищенного режима исполнения программ

- ▶ Возможность безопасной реализации межмодульного взаимодействия
- ▶ Адаптация библиотеки uClibc-ng под архитектуру Эльбрус
- ▶ Сценарий тестирования для этой библиотеки включает в себя тесты:
  1. Шифрование
  2. Работу с сетевыми протоколами
  3. Работу с потоками
  4. Проверку работоспособности функций стандартных библиотек языка C

# Autotest

- ▶ Утилита для тестирования ядра Linux. Autotest предоставляет окружение для запуска тестов.
- ▶ Механизм взаимодействия вида сервер-клиент. Процесс на сервере соединяется по протоколу ssh с удаленной машиной, запускает Autotest клиент, выполняет задачу на клиенте и собирает результаты из тестовой машины.
- ▶ Позволяет следить за выводом с консоли и может так же взаимодействовать с внешними службами, дающими доступ к журналам при падении системы, и даже перезагружать тестовые машины при наличии такой возможности.

# Существенные проблемы разработки

- ▶ Обработка установки нового ядра
- ▶ Запуск тестов защищенного режима исполнения программ в chroot окружении
- ▶ Хранение результатов тестирования
- ▶ Запуск тестов на всех доступных версиях ядра

# Обработка установки нового ядра

- ❑ Установка производится путем распаковывания deb пакетов
- ❑ Нет прописанных сценариев пре- или пост-инсталляции

Необходимо:

- Внести изменения в файл загрузки системы
- Обеспечить автоматическую загрузку всех необходимых для этой версии ядра модулей

## Решение

- Автоматическое извлечение параметров загрузки из файла `bootconf` для добавления новой метки
- Использование команды `dermod` с верно указанными ключами для реализации автоматической загрузки модулей нужной версии ядра



# Запуск тестов защищенного режима исполнения программ в chroot- окружении

- ❑ Для запуска внутреннего пакета тестирования библиотеки uClibc-ng нам требуется chroot-окружение с поддержкой как обычного, так и защищенного режима исполнения программ
- ❑ Необходимо монтирование не только системных директорий, но и директории с имеющимися тестами библиотеки uClibc-ng

## Решение

- ▶ Обеспечение копирования и извлечения новой версии тестов из смонтированной директории при каждом запуске системы тестирования
- ▶ Внесение правок об изменении файла autofs на тестовой машине

# Хранение результатов тестирования

Результаты тестирования библиотеки uClibc-ng защищенного режима исполнения программ должны храниться в наглядном виде в директориях, удобных для разработчиков.

## Решение

- ▶ Поиск файлов отчета тестирования, в которых была зафиксирована ошибка
- ▶ Отправка по почте оператору файла со списком тестов, завершившихся с ошибками
- ▶ Хранение результатов как на тестовой машине, так и на сервере

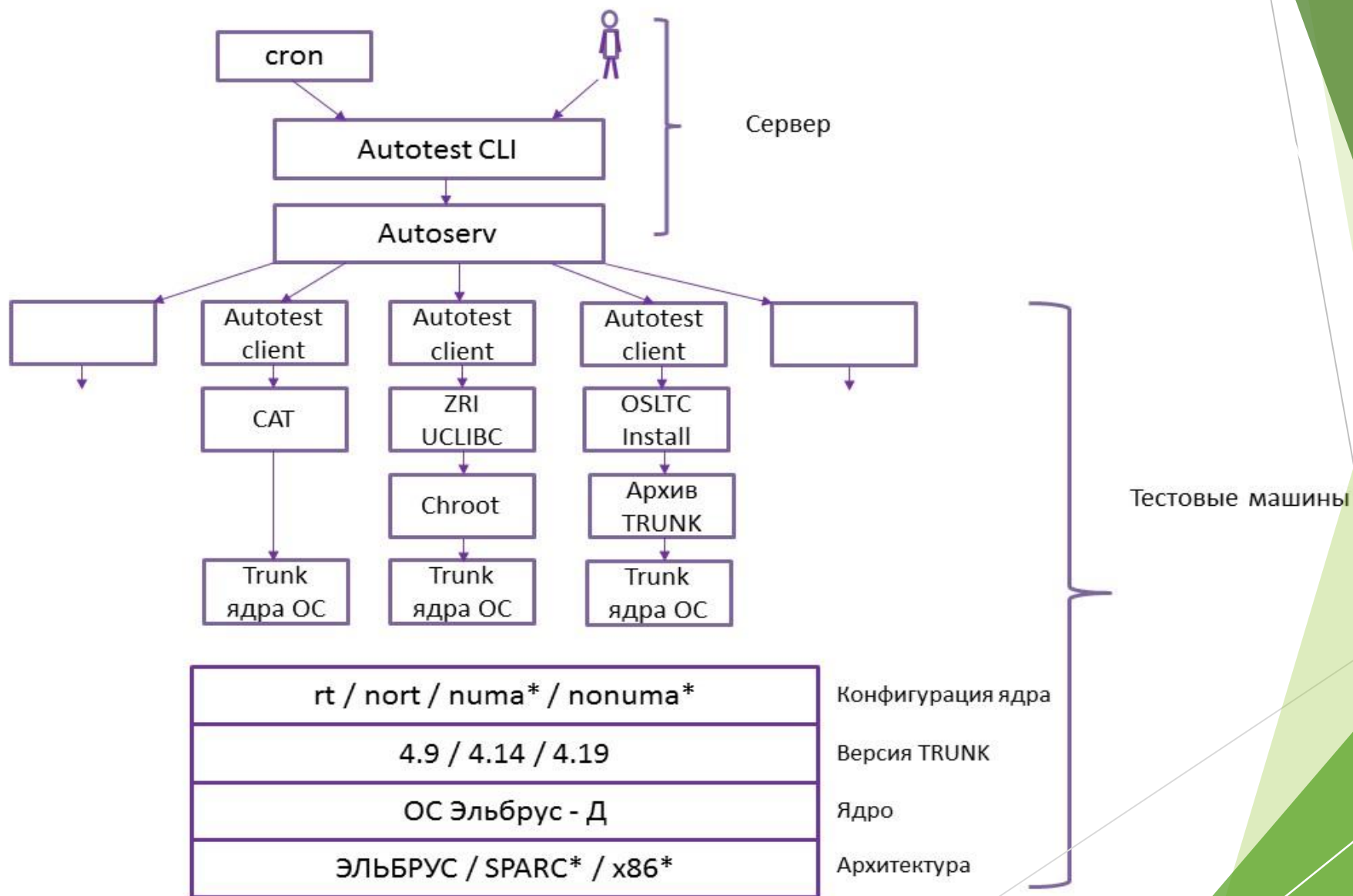
# Запуск тестов на всех доступных версиях ядра

Последние ревизии ядра ОС «Эльбрус-Д» имеют две доступные сборки, в том числе сборку с подсистемой реального времени (RT). Для обеспечения полного тестирования системы необходимо было запустить SAT на обеих сборках ядра.

## Решение

- ▶ Редактирования имеющегося файла загрузки системы bootconf
- ▶ Перезагрузка системы с новой версией ядра

# Схема системы тестирования



# Отчеты о работе системы

*Host = ed401-07*

*Build = 4.19.0-0-22020-e2s*

*Following errors were found:*

*/opt/mcst/check/STP/results/last/localhost/posixtestsuite/posixtestsuite\_1818  
0836.err*

*LTP-latest failed tests:*

*delete\_module01 delete\_module01*

*Delete\_module03 delete\_module03*

*conformance/interfaces/mmap/mmap\_6-2: execution: UNRESOLVED*

*conformance/interfaces/pthread\_create/pthread\_create\_3-2: execution: HUNG*

# Отчеты о работе системы

## ► Защищенный режим:

.... *inet/tst-res^MPASS inet/tst-res*

.... *inet/tst-sock-nonblock^MPASS inet/tst-sock-nonblock*

*Total skipped: 11*

*Total failed: 32*

*Total passed: 472*

## ► Обычный режим:

.... *locale-mbwc/tst\_wcstombs^MFAIL locale-mbwc/tst\_wcstombs got 1 expected 0*

.... *inet/tst-ifaddrs^MPASS inet/tst-ifaddrs*

*Total skipped: 11*

*Total failed: 28*

*Total passed: 476*

# Результаты

- ▶ Разработана и реализована система автоматического тестирования ядер ОС «Эльбрус-Д»
- ▶ Добавлена возможность включения в сценарии тестирования сторонних групп тестов
- ▶ Автоматизировано проведение тестирования как в обычном, так и в защищенном режимах исполнения программ
- ▶ Реализована возможность верификации разных сборок и разных версий ядра на базе архитектуры Эльбрус

**Спасибо за внимание**