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

Выпускная квалификационная работа магистра

Разработка универсальной библиотеки и программы тестирования устройств передачи данных

Студент Яндушкин Д. А., ФРТК, 913 гр.
Научный руководитель: д.т.н Семенихин С. В.

Библиотека тестирования устройств передачи данных
Требования к программному интерфейсу:

- Единая унифицированная структура данных для работы с устройствами передачи данных

- Стандартные элементы алгоритмов тестирования

- Функции для разработки тестовых алгоритмов

- Вывод и хранение результатов

Существующие программные решения для выполнения тестирования

STDP-TMS(ЗАО «МЦСТ»)
+ поддержка любых устройств, гибкая настройка, быстрое добавление новых тестов, единый вывод результатов
- у каждого теста свой набор параметров

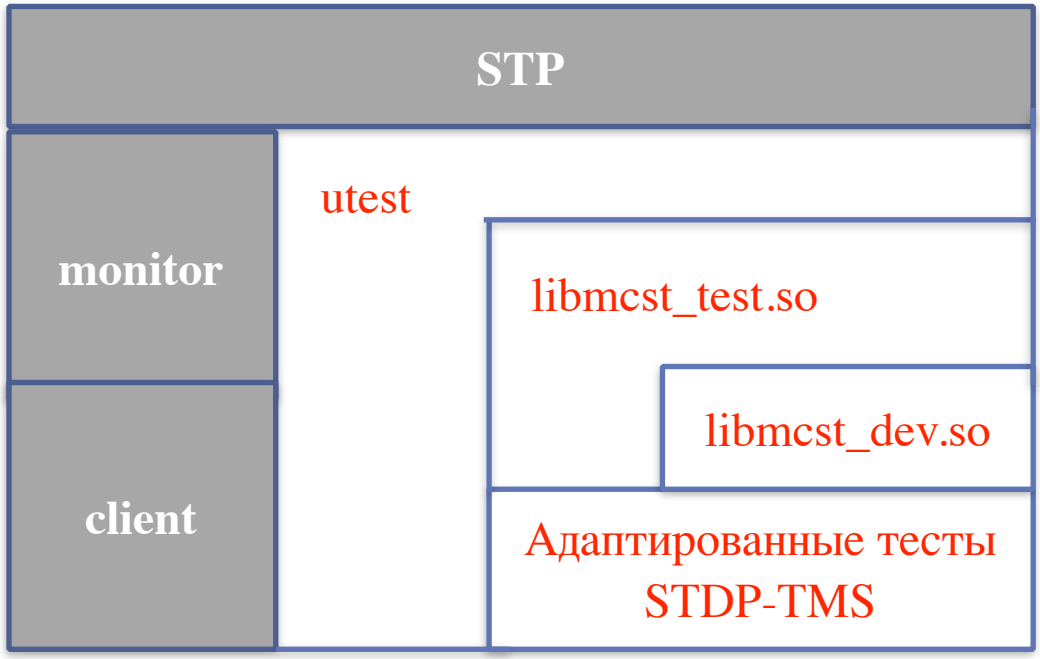
SunVTS (Sun Microsystems)
+ единый набор параметров для тестов, гибкая настройка, единый вывод результатов
- ограниченный набор поддерживаемых устройств, ограниченный набор ВК и ОС, отсутствует возможность добавления новых тестов

Специализированные программные решения
+ поддержка любых устройств
- произвольный набор параметров, не форматированный вывод результатов

Постановка задачи

Разработать универсальную тестовую систему для проверки устройств передачи данных.

Система должна содержать ограниченный набор тестовых алгоритмов и задаваемых параметров.



STP – система управления тестированием

Monitor – запуск набора тестов

Client - контроль за организацией и взаимодействием

utest – программа для запуска алгоритмов тестирования

Libmcst_test.so – библиотека тестирования

Адаптированные тесты STDP-TMS тесты для прочих устройств, собраны в динамические библиотеки

Libmcst_dev.so – библиотека для взаимодействия с устройства

Библиотека для работы с устройствами

- Универсальный программный интерфейс
- Унифицированный набор функций для всех устройств

Набор функций:

```
mdev_init() - Инициализация
mdev_open() - Получение файлового дескриптора
mdev_attr_set() - Установка атрибутов
mdev_attr_get() - Получение атрибутов
mdev_write() - Запись данных
mdev_read() - Считывание данных
mdev_poll() - Вызов poll()
mdev_ioctl() - Вызов ioctl()
mdev_ping() - Проверка приема данных
mdev_selftest() - Самопроверка
mdev_close() - Окончание работы
```

Библиотека тестирования устройств передачи данных

Единая унифицированная структура данных

Единая структура данных создана для облегчения работы с устройствами различного типа. Покрывает атрибуты, характеристики всех устройств и параметры тестирования

```
struct stp_dev_t {
    stp_attr_t attr;        // параметры устройства и теста
    stp_res_t res;          // результаты тестирования
}
```

Атрибуты:

```
struct stp_attr_t {
    int speed;
    int realtime;
    int affinity;
    char *device_path;
    …
}
```

Результаты:

```
struct stp_res_t {
        int write;
        int read;
        int success;
        int errors;
        …
}
```

Библиотека тестирования устройств передачи данных
Стандартные элементы алгоритмов тестирования

1) Последовательный

void write_read_data(stp_dev_t *dev)

Подготовка данных, запись в устройство, считывание с устройства, сравнение полученных и записанных данных

2) Одновременные запись и считывание

void write_read_duplex(stp_dev_t *dev)

Подготовка данных, одновременная запись и считывание данных с устройства, сравнение полученных и записанных данных

3) Последовательные запись и считывание, с использованием системного вызова poll()

void write_read_poll(stp_dev_t *dev)

Подготовка данных, запись в устройство, ожидание события с помощью системного вызова poll(), считывание данных с устройства, сравнение данных

Библиотека тестирования устройств передачи данных
Функции для разработки тестовых алгоритмов

Основной функционал	**Значение**
stp_mdev_read(stp_dev_t *dev)	Считывание данных
stp_mdev_write(stp_dev_t *dev)	Запись данных
algorithm_run_enable(stp_dev_t *dev, char *work_types)	Разрешение запуска алгоритма для устройств
algorithm_run_disable(stp_dev_t *dev, char *work_types)	Запрещение запуска алгоритма для устройств
stp_mdev_open(stp_dev_t *dev)	Получение файлового дескриптора
stp_mdev_close(stp_dev_t	Закрытие файлового дескриптора
print_result(stp_dev_t *dev)	Вывод результатов

Библиотека тестирования устройств передачи данных
Пример алгоритма тестирования

```
char *work_types[] = {"CPU", "MEM", NULL};       // разрешенные устройства
char *notwork_types[] = {“ETH", “COM", NULL};  // запрещенные устройства

void example_test(stp_dev_t *dev)
{
        start_alg_time(dev);          // Начало отсчета времени
        algorithm_run_enable(dev, work_types);
        algorithm_run_disable(dev, notwork_types);
        while (number != 0 || test_errors != 0 || time != 0 ) {
                body_example_test(dev);  // Алгоритм тестирования
                dev->pass.n++;                // Количество проходов
        }
        print_example_test(dev);   // Вывод результатов
}

void body_example_test(stp_dev_t *dev) {
        stp_mdev_write(dev);       // запись данных в устройство
        stp_mdev_read(dev);        // считывание данных с устройств
}
```

Адаптация и доработка тестов STDP-TMS

Перенесены тесты для следующих устройств:

– **CPU** (дополнительно реализованы алгоритмы параллельного и последовательного запуска всех тестов для этого устройства)
– **Memory**
– **Sound**
– **Video**
– **Hdd**
– **Monitor**
– **Mouse**
– **Keyboard**

Тесты для каждого устройства, собраны в динамические библиотеки.

Универсальная программа для запуска тестов
Требования:

Взаимодействие с библиотекой тестирования
С помощью программного интерфейса библиотеки тестирования устройств передачи данных, вести тестирование

Обработка параметров
- Параметры, определяющие ход работы теста (время, количество ошибок и прочие)
- Параметры, описывающие атрибуты устройства (тип устройства, скорость передачи данных, …)

Запуск тестирования
Запуск стандартных алгоритмов тестирования и внешних алгоритмов тестирования

Набор параметров и Примеры запуска тестирования

```
utest [-h -H|-t|-T|-d|-n|-p|-e|-s|-c|-S|-v|-
P|-I|-A|-r|-a|-type|-time|-number|-dev|-
porosity|-errors|-server|-silent|-verbose|-
debug|-normal|-client|-alg|-ip_adress|-port|-
poll_event|-writetimeout|-readtimeout|-
polltimeout|-ioctltimeout|-timeout|-
micro_porosity|-connect_timeout|-size|-
index_msg|-rus|-eng|-affinity|-realtime|-
logoff|-time_info|-silent_res]
```

./utest --type ETH -I 127.0.0.1 -client —alg write_read_data --verbose -n 15 -t 10 -e 2 --index 1 --size 8 -r -a 2 --timeout 1

./utest --type COM -d /dev/ttyS1 --alg write_read_data --verbose -n 15 -t 10 -e 2 --index 1 —size 8 -r -a 2 —timeout 1

Результаты:

- Разработана библиотека тестирования устройств передачи данных с ограниченным набором тестовых алгоритмов и задаваемых параметров
- Разработана программа запуска тестов
- Адаптированы некоторые тесты STDP-TMS
- Работа системы была проверена на устройствах: COM, ETH, MMRM, MMR, RDMA, MOP, MPV
CPU, memory, video, sound, keyboard, mouse, hdd, monitor

Перспективы:

- Сопровождение программного продукта
- Использование библиотеки тестирования и программы запуска тестов для проверки устройств вычислительных комплексов в компании ЗАО «МЦСТ»

Спасибо за внимание!