***С.В. Клюев аспир.; рук. О.В. Бартеньев к.т.н., доц.,
(НИУ «МЭИ», г. Москва)***

**ЗАДАЧИ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ
ОПЕРАЦИОННОЙ СИСТЕМЫ Z/OS**

## Введение

Операционная система (ОС) z/OS, разработанная компанией IBM, предназначается для поддержки архитектуры z/Architecture, которая используется в мейнфреймах на платформе zSeries. Структура и заложенные принципы ОС z/OS отличны от распространенных в настоящее время UNIX-подобных ОС. На данный момент актуальной задачей является автоматизация тестирования данной ОС, которая подтверждается повседневной практикой. Очевидно, что построение системы автоматизации тестирования z/OS невозможно без её детального изучения. В работе рассматривается организация одного из библиотечных наборов данных – PDSE (Partitioned Data Set Extended), формулируются проблемы его тестирования, описывается текущая технология тестирования PDSE, формулируются задачи автоматизации тестирования и предлагается структура системы автоматизации тестирования z/OS.

## Организация PDSE наборов данных

PDSE наборы данных используются для хранения исходных текстов программ, процедур, заданий, текстовых документов и т.п. При создании PDSE-набора данных область внешней памяти разбивается на блоки по 4 КБ, именуемые страницами, в которых содержатся блоки оглавления и разделов (данных). Для обеспечения динамического расширения оглавления набора данных, а также изменения параметров разделов, блоки оглавления распределяются разрывным образом вместе с блоками данных.

Оглавление имеет индексную организацию, состоящую из AD (Attribute Directory) и ND (Name directory) страниц, которые обеспечивают взаимосвязь со страницами разделов. Иерархическая структура AD-страниц, реализована по принципу В-дерева и содержит информацию об атрибутах разделов (время создания, время последнего изменения, информацию о размере и т. д.), а также о самом наборе данных.

Разделы PDSE отслеживаются посредством идентификатора FSN (files sequence number). Указатель на страницу, содержащую данные раздела, хранится в FPM (fragment parcel map). ND-страницы так же организованы по принципу В-дерева и связывают индивидуальные имена разделов с их идентификаторами FSN. Внутренняя структура элементов оглавления представлена на рис. 1 [1].

 1 - AD страница

 2 - ND страница

 3 - Страницы разделов

Рис. 1. Внутренняя структура элементов оглавления

## Задачи и средства тестирования PDSE наборов данных

Важными задачами тестирования PDSE наборов является проверка целостности и доступности данных. Можно выделить следующие ключевые моменты в решении этих задач:

* Проверка целостности структуры набора данных.
* Тестирование поведения набора данных в условиях совместной обработки его несколькими системами, объединёнными в кластер.
* Оценка производительности, связанная с проверкой скорости выполнения таких операций, как добавление, удаление и поиск раздела в наборе данных.
* Проверка совместимости версий.
* Тестирование работы в «нештатных» условиях.

Средства ОС z/OS предоставляют ряд утилит и сервисов используемых при решении поставленных задач. Средства проверки целостности данных, представлены набором внутренних команд и утилит z/OS [1], которые позволяют идентифицировать проблемы, связанные с разделением ресурсов и вовлечённые в них PDSE наборы данных, а также верифицировать структуру директории. Производительность системы, в частности скорость обработки наборов данных, осуществляется средствами Resource Measurement Facility (RMF), позволяющей отслеживать скорость обработки данных z/OS или кластера z/OS и имеет инструменты для оптимизации параметров и конфигурации системы (z/OS или кластера z/OS).

Описанные средства предоставляют возможности по диагностированию проблемы, но в большинстве случаев требуют предварительного ручного анализа условий возникновения неисправности, что не применимо в процессе исполнения группы автоматизированных тестов. ОС z/OS не предоставляет специализированных средств имитации «нештатных» условий, таких, как эмуляция нехватки памяти, пространства для создания набора данных, задание некорректных параметров в процессе создания и др. Такие ситуации могут быть воссозданы при помощи языка управления заданиями JCL и языков программирования, таких как REXX и HLASM.

## Задачи автоматизации тестирования

В процессе жизненного цикла большого программного продукта, такого, как серверная ОС, при добавлении и модификации функционала ОС, происходит неизбежное изменение исходного кода. При этом критически важной является проверка корректности работы ранее реализованного функционала, которая возложена на регрессионные тесты. За годы развития ОС количество тестов в регрессионной базе достигает более десятка тысячи экземпляров, выполнение которых является очень ресурсоёмким процессом и несёт большие временные затраты. Поэтому после внесения изменения в исходный код системы возникают задачи автоматизации тестирования, главными из которых являются следующие:

* Разработка базы данных тестов и ошибок, обнаруженных в процессе эксплуатации z/OS, и системы управления базой данных и системы анализа накопленных данных.
* Выбор средствами системы анализа данных подмножества регрессионных тестов, для проверки корректности работы измененных частей z/OS.
* Определение тестов, подлежащих модификации в связи с изменением тестируемых параметров.
* Создание новых тестов, увеличивающих тестовое покрытие на основе анализа существующих тестов.

Ввиду закрытости исходного кода z/OS решение поставленных задач подразумевает создание системы автоматизации тестирования (рис. 2), использующей заранее описанные требования, заданные в технической документации.



 1 - Среда выполнения тестов

 2 - Виртуальная машина

Рис. 2. Структура системы автоматизации тестирования

Анализатор документации осуществляет синтаксический разбор и извлечение параметров, которые будут использоваться в конструкторе, содержащем модели компонентов ОС, таких, как организация наборов данных, методы доступа, подсистема ввода-вывода и прочих подсистем.

Конструктор, применяя комбинаторные техники (pairwise testing), анализ граничных значений (boundary value analysis) и разбиение на эквивалентные множества (equivalent partition), осуществляет формирование исходного кода тестов, учитывая информацию о существующем тестовом покрытии, полученную из базы данных тестов.

Среда выполнения тестов основана на технологии виртуальных машин [2], ключевым компонентом которой является планировщик, реализующий распределение тестов и эффективный алгоритм загрузки виртуальных машин.

## Заключение

В работе рассмотрены структура и особенности средств тестирования набора данных PDSE, сформулированы задачи автоматизации тестирования и предложена структура автоматизированной системы тестирования z/OS. Цель автоматизации - снижение трудоемкости тестирования и повышение его эффективности.

## Библиографический список

1. Partitioned Data Set Extended Usage Guide, IBM, SG24-6106-01.

2. Matthew Kaplan, Tracy Kimbrel, Kevin Mckenzie, Richard Prewitt, Maxim Sviridenko, Clay Williams and Cemal Yilmaz. Test Machine Scheduling and Optimization for z/OS.Honolulu, Hawaii: In the Proceedings of the IEEE Symposium on Computational Intelligence in Scheduling (CISched '07), April 1-5, 2007. стр. 27-24.