Btparser

A program failure analysis library

Karel Klíč

November 30, 2012

Contents

Chapter 1

Overview

Failures of computer programs are omnipresent in the information technology industry: they occur during software development, software testing, and also in production. Failures occur in programs from all levels of the system stack. The program environment differ substantially between kernel space, user space programs written in C or C++, Python scripts, and Java applications, but the general structure of failures is surprisingly similar between the mentioned environments due to imperative nature of the languages and common concepts such as procedures, objects, exceptions.

Btparser is a collection of low-level algorithms for program failure processing, analysis, and reporting supporting kernel space, user space, Python, and Java programs. Considering failure processing, it allows to parse failure description from various sources such as GDB-created stack traces, Python stack traces with a description of uncaught exception, and kernel oops message. Infromation can also be extracted from the core dumps of unexpectedly terminated user space processes and from the machine executable code of binaries. Considering failure analysis, the stack traces of failed processes can be normalized, trimmed, and compared. Clusters of similar stack traces can be calculated. In multi-threaded stack traces, the threads that caused the failure can be discovered. Considering failure reporting, the library can generate a failure report in a well-specified format, and the report can be sent to a remote machine.

Due to the low-level nature of the library and implementors' use cases, most of its functionality is currently limited to Linux-based operating systems using ELF binaries. The library can be extended to support Microsoft Windows and OS X platforms without changing its design, but dedicated engineering effort would be required to accomplish that.

Part I

Concepts

Chapter 2

Stack Trace Normalization

Chapter 3

Stack Trace Clustering

Chapter 4

Core Dump Failure Analysis

Chapter 5

Wishlist

Security Impact.

ABI compatibility check.

Collecting environment data.

Part II

Implementation

Chapter 6

Overview

Btparser is implemented in the C language as defined in the C99 standard (ISO/IEC 9899:1999). It uses the C standard library and some additional libraries. No additional library is mandatory, though. When a library is not found by the build configuration script, the features requiring that library become unavailable. This approach improves both usability and portability of the library.

doxygen/refman

Chapter 7

Known Bugs

Empty.

Chapter 8

Wishlist

Stack trace for kerneloopses, Python, and Java.