



Advanced Methodologies for Memory Leak Detection in Software Programs

Faculty Name : Dr. Jyothi Shetty
Department of Computer Science and Engineering
R .V. College of Engineering, Bangalore-59

1.Problem Statement

Create a memory leak detection tool to identify and report unallocated memory instances in software. The tool should analyze runtime memory usage, pinpoint source code locations of leaks, and provide a detailed report. It must be efficient, cross-platform, and support various programming languages. Additionally, include comparative documentation among various tools and examples.

2) LITERATURE SURVEY

PAPER TITLE	DATE	INFERENCE
Memory Leak Detection Tools-A Comparative Analysis	6th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), August 27th & 28th 2021	<ul style="list-style-type: none">Memory leak is a serious problem in embedded systems as they are memory constrained devices.The results show that Mtrace tool takes maximum time for analysis and Electric fence tools takes minimum time for analysis.
Memory leak detection in Plumbr	Vladimir Šor, Satish Narayana Srirama and Nikita Salnikov-Tarnovski DOI: 10.1002/spe.2275	<ul style="list-style-type: none">Platforms like the JVM that have automatic memory management are typically thought to be memory leak-free.As an alternative, this paper discusses the statistical strategy for memory leak detection along with a commercial tool based on the technique called Plumbr.Afterwards, the tool is examined using three case studies of actual applications, and the results have been analysed

3) OBJECTIVES

- Prevent Resource Exhaustion
- Enhance System Stability
- Ease of debugging Memory related errors
- Mitigate Security Risks
- Promote Long-Term Reliability:

4) APPLICATIONS

- Software Development industry
- Finance and Banking
- Healthcare
- Telecommunications
- Energy and Utilities
- E-commerce and Retail
- Education and Research

5)TOOLS USED TO DETECT MEMORY LEAK

- Valgrind
- GDB (GNU Debugger)
- LeakSanitizer (AddressSanitizer)
- mtrace
- Memleax
- Collecting core dump using abrt and abrt-addon-ccpp
- Memory leak using default Linux tools such as sar, vmstat, pmap and meminfo.
- Electric fence

6)VALGRIND AND GDB DEBUGGER

Valgrind-

It is a memory leak detection tool used to check if there are any untoward memory allocations, or leaks in the program.

GDB Debugger-

It is a command line debugger which helps us to decode where the error might be occurring in the program

6)ELECTRIC FENCE,ABRT AND SAR COMMAND

ELECTRIC FENCE

- Software that overruns the boundaries of a malloc()memory allocation,
- Software that touches a memory allocation that has been released by free().

ABRT

- ABRT (Automatic Bug Reporting Tool), is a set of tools that help users detect and analyze application crashes.
- ABRT consists of the abrt daemon and a number of system services and utilities for processing, analyzing, and reporting detected problems.

SAR COMMAND

- The sar command in Linux (System Activity Reporter) can be utilized to monitor memory usage over time, which may indirectly help in detecting memory leaks by observing abnormal memory consumption patterns

7) LEAK SANITIZER,MTRACE AND

LEAK SANITIZER

- Leak Sanitizer (LSAN) is a tool used for detecting memory leaks in C and C++ programs. It is part of the AddressSanitizer suite, which is a memory error detector developed by Google.

MTRACE

- mtrace is a memory debugging tool available on Unix-like systems, particularly popular in Linux environments. It is useful in tracking dynamic memory allocation and allows developers to detect memory leaks and diagnose memory-related issues.

MEMLEAK

- Memleax is a lightweight memory debugging tool designed to detect memory leaks in C and C++ programs. Unlike some other memory debugging tools, Memleax focuses specifically on identifying memory leaks, making it a specialized tool for this purpose

8)CODE IMPLEMENTATION

This system replaces standard memory functions in C programs with custom versions (MyMalloc, MyCalloc, MyFree) to track memory allocation and deallocation. It records size, file, and line number information for each allocation, aiding in debugging and memory leak detection. Implemented through findLeak.c and findLeak.h.

9) API'S AND SYSTEM CALLS USED

- **malloc:** Allocates memory dynamically.
- **calloc:** Allocates memory for an array of elements, initializing them to zero.
- **free:** Deallocates previously allocated memory.
- **strcpy:** Copies a string.
- **memset:** Fills a block of memory with a particular value.
- **strncpy:** Copies a certain amount of characters from one string to another.
- **sprintf:** Prints formatted output to a string.
- **fopen:** Opens a file.
- **fwrite:** Writes data to a file.
- **fclose:** Closes a file.

10) CONCLUSION

Through this project we've realized the research done on this topic was lacking and hence it becomes an absolute necessity for us to continue research on this. We have learnt the usage of multiple tools, their inner workings and how it can make the life of a software developer easier, and help him to build robust, useful and memory efficient applications. A comparative study would be helpful for a person to choose the best suited tool for his requirement.

TEAM MEMBERS :

- | | |
|-----------------|--------------|
| 1. PRAMATH K P | (1RV22CS142) |
| 2. ROHIT SURESH | (1RV22CS163) |