Assignment_5-09-2024

1. Explore memory leakage utilities in linux

1.1 GDB (GNU Debugger)

- Description: While primarily a debugging tool, GDB can help track memory allocations, deallocations, and potential leaks by inspecting the program's memory state.
- Usage

```
gdb ./program_file
(gdb) run
(gdb) info proc mappings
```

  It allows to explore the memory mappings of the process and track down memory-related issues.

1.2 AddressSanitizer (ASan)

- **Description**: A fast memory error detector that is part of the GCC and Clang compilers. It detects out-of-bounds accesses, use-after-free errors, and memory leaks.
- **Usage**

```
gcc -fsanitize=address -g program_file.c -o program_file
./program_file
```

1.3 LeakSanitizer

- **Description**: A memory leak detector that can be integrated with ASan or used as a standalone tool.
- **Usage**:

```
gcc -fsanitize=leak program_file.c -o program_file
./program_file
```

1.4 Smem

- **Description**: A simple utility to report memory usage per process, helping track memory leaks over time.
- **Usage**:

```
smem
```

2. Explore performance benchmarking tool

https://www.phoronix-test-suite.com/

Performance benchmarking tools help measure and evaluate the speed, efficiency, and overall performance of systems, applications, or code. There are many benchmarking tools in Linux designed for different purposes such as CPU, memory, disk, and network performance.

Use Cases:

- CPU Performance: sysbench, perf, stress-ng
- Memory Performance: sysbench, stress-ng
- Disk I/O Performance: fio, hdparm, sysbench
- Network Performance: iperf, lmbench
- Overall System Stress: stress-ng, Phoronix Test Suite

1. time

- **Description**: A basic Linux utility used to measure the execution time of a program (real, user, and system time).

- Usage: time ./program_file

Example Output:

```
real    0m0.002s
user    0m0.001s
sys     0m0.001s
```

2. perf

- **Description**: A powerful performance analysis tool that provides detailed information about CPU performance, cache usage, and more.
- **Usage:** perf stat ./program_file

Features:

- CPU cycles, cache hits, cache misses, instructions per cycle, etc.
- Profiling and tracing with perf record and perf report.

3. hyperfine

- **Description**: A command-line benchmarking tool to compare the performance of multiple commands. It runs the commands multiple times and reports the average execution time.
- **Usage:** hyperfine "./your_program"

4. sysbench

- **Description**: A versatile benchmarking tool that can test CPU, memory, disk I/O, and database performance.
- **Usage**
  1. CPU Benchmark:sysbench --test=cpu --cpu-max-prime=20000 run
  2. Memory Benchmark: sysbench --test=memory run
  3. Disk I/O Benchmark: sysbench --test=fileio --file-test-mode=seqread run

5. fio

- **Description**: A flexible and powerful disk I/O performance benchmarking tool.
- **Usage:** fio --name=randreadwrite --rw=randrw --size=1G --numjobs=4 --time_based --runtime=60s