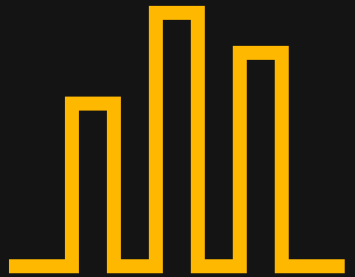
SYSTEM CALL PROFILING AND PERFORMANCE ANALYSIS TOOL.

Team Name = YaAm
Name mem 1 = Amit Shinde
Name mem 2 = Yash Singh Bhadauria



Purpose: Trace and analyze system calls made by a program.

Captures:
Number of calls for each system call.
Total execution time of each system call.
Any failed system calls.

Benefits:
Identifies performance bottlenecks.
Helps debug failed system calls.
Monitors system resource usage.

Output:
Summary report displayed in the console or saved to a text file.

Use Cases: Performance optimization and error debugging.

Implemetation

Run strace:
Executes the target program with strace to capture system calls (-T for time, -e trace=all to capture all system calls).
Captures the output from stderr, which contains system call details.

Regex Parsing:
Uses a regular expression to match system calls and their execution time.
Extracts the system call name (\w+) and the time spent (\+(\d+\.\d+)).

Store Data:
Tracks the count of each system call.
Accumulates the total time spent on each system call.

Return Results:
Returns a dictionary where the key is the system call name and the value is a dictionary containing the count and total time.

Use Case

Performance Optimization:
Identifies time-consuming system calls (e.g., read, openat).
Helps optimize resource-heavy operations.

Error Debugging:
Tracks failed system calls for quick error diagnosis.
Example: Identifies failed openat calls due to missing files.

Resource Monitoring:
Monitors system call frequency and time.
Detects high resource usage (e.g., memory or I/O).

Program Behavior Comparison:
Compares system call stats between different programs.
Helps identify more efficient implementations.

Detailed System Call Breakdown:
Provides insights into the time spent on each system call.
Enables targeted optimization of slow calls.

Output Format

It gives output in the following format:

- Name of stystem call
- Number of times itwas called
- Toltal time taken by system calls

amit.txt

```
System Call Summary:
execve: 1 calls, 0.004155 s
openat: 2 calls, 0.000221 s
newfstatat: 3 calls, 0.000427 s
close: 2 calls, 0.000177 s
read: 1 calls, 0.000154 s
pread64: 4 calls, 0.000398 s
mprotect: 4 calls, 0.000217 s
arch_prctl: 1 calls, 0.000041 s
set_tid_address: 1 calls, 0.000035 s
set_robust_list: 1 calls, 0.000036 s
rseq: 1 calls, 0.000035 s
prlimit64: 1 calls, 0.000037 s
munmap: 1 calls, 0.000047 s
getrandom: 1 calls, 0.000065 s
write: 2 calls, 0.000071 s

```

THANK YOU

8983235362
yash.bhadauria.22031@iitgoa.ac.in