Contents

[Project phases 2](#_Toc7278)

[Phases Explanation 2](#_Toc7279)

[Literature Understanding 2](#_Toc7280)

[Code Skeleton 2](#_Toc7281)

[Detection Mechanism For Locking System Calls 2](#_Toc7282)

[Printing stacktrace 2](#_Toc7283)

[Locked Resource Detection 3](#_Toc7284)

[Starving Process Detection 3](#_Toc7285)

[Detection of lock type that is creating issues 3](#_Toc7286)

[Testing 3](#_Toc7287)

[Plan of Action 3](#_Toc7288)

[Semester One 3](#_Toc7289)

[Phase-1: Literature Understanding. 3](#_Toc7290)

[Phase 2: Building Prototype 3](#_Toc7291)

[Semester Two 3](#_Toc7292)

[*Post Demo Plan.* 4](#_Toc7293)

# Project phases

1. Literature Understanding.
2. Code Skeleton -tracing grandchild(threads) and keeping their records in specific data structure.
3. Specific calls that uses for Lock detection.
4. Adding Libunwind functionality to pinpoint problematic methods using stack trace.
5. Detection of locked resources.
6. Detection of waiting process.
7. Notifying the type of lock.
8. TESTING.

# Phases Explanation

## Literature Understanding

* Ptrace (API)
* Inter Process Communication Understanding –Signal Handling.
* Linux: Debugging and Performance Tuning (Book)
* “C Programming with Linux" (Seven Courses on edx)
* Self Service Linux (Book)

## Code Skeleton

1. Working on practice exercises to understand and learn ptrace
2. Understanding strace code for structural understanding
3. Writing Mini strace
4. Evolving code into skeleton code for project

## Detection Mechanism For Locking System Calls

*?*

*----And Disrupting them after detection... Maybe???*

## Printing stacktrace

1. Learning libunwind library
2. Adding functionality in skeleton
3. Testing

## Locked Resource Detection

?

## Starving Process Detection

?

## Detection of lock type that is creating issues

?

## Testing

* Self-written Small Test Cases that have the issue built in.
* Testing of some of the well-known software.

# Plan of Action

## Semester One

### Phase-1: Literature Understanding.

Status: Complete

### Phase 2: Building Prototype

Status: Complete

#### Functionality

* Tracing system call of a process.
* Tracing system call of the forked process.
* Can start a tracee via prototype
* Can connect to the tracee via prototype

## Semester Two

|  |  |  |
| --- | --- | --- |
| Week **1** | **1** – 8 Feb | Testing and Organizing Prototype. |
| Week **2** | **8** – 15 Feb | Finding specific calls that use for locking mechanism. |
| Week **3** | **15** – 22 Feb | Adding Libunwind functionality. |
|  | | |
| Week **4** | **22** – 1 March | Testing stack trace functionality. |
| Week **5** | **1** – 8 March | Detection of locked resources. |
| Week **6** | **8** – 15 March | // |
| Week **7** | **15** – 22 March | Detection of waiting process. |
| Week **8** | **22** – 29 March | // |
|  | | |
| Week **9** | **29** – 5 April | Notifying the type of lock. |
| Week **10** | **5** – 12 April | Pre-Demo Testing |
| Week **11** | **12** – 19 April | Finishing and Demo Planning. |

### *Post Demo Plan.*

|  |  |
| --- | --- |
| **April 24 – May 24** | Post-Demo Testing and Debugging. |
| Project Documentation. |
| Releasing Beta Version. |