All rights reserved. No part of this document shall be reproduced, stored in a retrieval system, or transmitted by any means, electronics, mechanical, photocopying, recording, or otherwise, without written permission from the authors. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

Trademark

All trademarks, trade names, or logos mentioned or used are the property of their respective owners. Every effort has been made to properly capitalize, punctuate, identify, and attribute trademarks and trade names to their respective owners.
## Team Roles

<table>
<thead>
<tr>
<th>Name</th>
<th>Phase 1</th>
<th>Phase 2</th>
<th>Phase 3</th>
<th>Phase 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rachel Chavez</td>
<td><strong>Leader</strong> - Business Analyst</td>
<td>Time Keeper - System Arch.</td>
<td>Software Dev.</td>
<td>Minute Taker - Tester</td>
</tr>
<tr>
<td>Eduardo Castillo</td>
<td>Project Analyst</td>
<td>Minute Taker - System Analyst</td>
<td>Software Engineer</td>
<td><strong>Leader</strong> - Tester</td>
</tr>
<tr>
<td>Claudia De Leon</td>
<td>Minute Taker - Functional Analyst</td>
<td>Requirement Validation</td>
<td><strong>Leader</strong> - GUI Designer</td>
<td>Time Keeper - Quality Assurance</td>
</tr>
<tr>
<td>Carlos Lacasa</td>
<td>Time Keeper - Solutions Arch.</td>
<td><strong>Leader</strong> - Testing Analyst</td>
<td>Minute Taker - Dev. Manager</td>
<td>Software Dev. - Tester</td>
</tr>
<tr>
<td>Gerardo Guijarro</td>
<td>Software Dev.</td>
<td>Software Analyst</td>
<td>Time Keeper - Quality Assurance</td>
<td>Software Dev. - Tester</td>
</tr>
</tbody>
</table>
Project Management

Deliverable 1

High Level User Requirements
Determine Personnel Organization
Identify Software and Hardware...
Identify Tasks, Milestones, Deliverables
Documentation – Deliverable 1
Presentation – Deliverable 1

Deliverable 2

Identify Functional Req....
Identify Non-functional ...
Formulate Use Cases
Create Use Case Diagrams
Documentation-Deliverable 2
Presentation – Deliverable 2
Project Management

Deliverable 3

<table>
<thead>
<tr>
<th>Date</th>
<th>Deliverable 3</th>
<th>High-level System Archit.</th>
<th>System Decomposition</th>
<th>Design GUI</th>
<th>Code Generation</th>
<th>Doc. – Deliverable 3</th>
<th>Pres. – Deliverable 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6/10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6/20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6/30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7/10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7/20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Deliverable 4

<table>
<thead>
<tr>
<th>Date</th>
<th>Deliverable 4</th>
<th>Complete Implementation</th>
<th>Debugging</th>
<th>System Evaluation</th>
<th>Doc. - Deliverable 4</th>
<th>Pres. – Deliverable 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/1/2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/11/2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/21/2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/31/2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7/10/2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7/20/2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7/30/2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8/9/2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Start: Blue
Complete: Red
Problem Def. & Limitations

Moodle
• Personalized educational experience
• Access to course materials

vMoodle
• Virtual Machines as valuable tools for portable educational modules
• Easily replicated and deployed

vMoodle Mobile
- Fast data access
- Mobility
- Increased popularity

vMoodle Limitation
- Does not take advantage of the increasing popularity of mobile devices therefore, it restricts users to use PCs.
High Level System Objectives

The system shall:

• Provide the similar functionality to vMoodle
• Provide fast access to data
• Be responsive
• Make efficient use of the device screen.
• Have intuitive UI and self-explanatory UI elements.
• Provide error messages when necessary.
System Deployment

**Hardware:**
- Android Device
  - 800MHz single core processor
  - 196MB of RAM
  - 3.5-inch HVGA touchscreen display
  - 3G or Wi-Fi
- **Software:**
  - Android 2.3 Gingerbread

**Hardware:**
- vMoodle Server
  - vMoodle Mobile - User Interface
  - vMoodle Mobile - Logic Subsystem
  - vMoodle Mobile - Storage Tier
  - Apache Server
  - MySQL DB

**Cloud**
- VirtualBox for Linux version: 4.1.18

**Network Speed:** 446 kbps

**Hardware:**
- Cloud
  - Apache 2.2.21 (Ubuntu)
  - PHP 5.3.8

**Hardware:**
- vMoodle Server
  - MySQL DB

**Software:**
- Linux: Ubuntu 8.1

**Software:**
- VirtualBox
  - Ubuntu 8.1

**Hardware:**
- 1 TB of disk space
- 4 GB of RAM
- 2 GHz processor
username
password
domain

Create Virtual Machine

Lab 1: Adding a system call to the Linux Kernel

In this project you will study the system-call interface provided by the Linux operating system and learn how user programs communicate with the operating system kernel via this interface. Your task is to incorporate a new system call into the kernel thereby expanding the functionality of the operating system. Part 1: Getting Started: A user-mode procedure call is performed by passing arguments to the called procedure either on the stack or through registers, saving the current state and the value of the program counter, and jumping to the beginning of the code corresponding to the called procedure. The process continues to have the same privileges as before. System calls

Grade: - / 100
Due Date: 9/30/2012
Create Virtual Machine
Lab 1: Adding a system call to the Linux Kernel

In this project, you will study the system-call interface provided by the Linux operating system and learn how user programs communicate with the operating system kernel via this interface. Your task is to incorporate a new system call into the kernel thereby expanding the functionality of the operating system.

Getting Started:
- A user-mode procedure call is performed by passing arguments to the called procedure either on the stack or through registers, saving the current state and the value of the program counter, and jumping to the beginning of the code corresponding to the called procedure.
- The process continues to have the same privileges as before.

System calls
System Design (Instructor Path)

Virtual Machine Templates

Assignments

Lab 1: Adding a system call to the Linux Kernel

In this project you will study the system-call interface provided by the Linux operating system and learn how user programs communicate with the operating system kernel through this interface. Your task is to incorporate a new system call into the kernel thereby expanding the functionality of the operating system.

Part 1: Getting Started:
A user-mode procedure call is performed by passing arguments to the called procedure either on the stack or through registers, saving the current state and the value of the program counter and...

Due Date: 9/30/2012

Modify Assignment

View Virtual Machine
Lab 1: Adding a system call to the Linux Kernel

In this project you will study the system-call interface provided by the Linux operating system and learn how user programs communicate with the operating system kernel via this interface. Your task is to incorporate a new system call into the kernel thereby expanding the functionality of the operating system.

Part 1: Getting Started
A user-mode procedure call is performed by passing arguments to the called procedure either on the stack or through registers, saving the current state and the value of the program counter, and jumping...
```java
private JSONObject parserObj;

public JSONParser(String response) {
    try {
        parserObj = new JSONObject(response);
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

public String getUserId() {
    return parserObj.optString("user_id");
}

public boolean isValid() {
    if (this.parserObj == null)
        return false;

    if (parserObj.optString("isValid").equalsIgnoreCase("TRUE"))
        return true;
    return false;
}
```
System Implementation

Web Service
- login - startVM
- refresh - stopVM
- createAssignment - pauseVM
- submitAssignment - getResources
- createVM - modifyVM
- refreshVM - deleteVM

Storage

- Installed on Visa lab private cluster
- Allows additional guest operating systems

MySQL (RDBMS)
# System Evaluation

## Experiment Testbed

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Device</th>
<th>RAM</th>
<th>CPU</th>
<th>Hard Disk</th>
<th>Network Speed</th>
<th>Screen Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cellphone</td>
<td>512 MiB</td>
<td>800 MHz</td>
<td>125 MB</td>
<td>446 kbps</td>
<td>3.5 inch</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>4 GB</td>
<td>2 x 2.2 GHz</td>
<td>320 GB</td>
<td>11 Mps Wi-Fi (fiu-scs)</td>
<td>N/A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software</th>
<th>Device</th>
<th>OS</th>
<th>Additional Soft.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cellphone</td>
<td>Android 2.3 Gingerbread</td>
<td>ConnectBot</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>Windows 7 Professional</td>
<td>N/A</td>
</tr>
</tbody>
</table>
System Evaluation

Use Cases Performance Using Wi-Fi

<table>
<thead>
<tr>
<th>Use Cases</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>startVM</td>
<td>2.0 ± 0.2</td>
</tr>
<tr>
<td>stopVM</td>
<td>1.5 ± 0.1</td>
</tr>
<tr>
<td>pauseVM</td>
<td>1.6 ± 0.2</td>
</tr>
<tr>
<td>createVM</td>
<td>3.5 ± 0.3</td>
</tr>
<tr>
<td>deleteVM</td>
<td>1.8 ± 0.2</td>
</tr>
</tbody>
</table>

- **phone w. wifi**
- **pc w. wifi**
System Evaluation

Use Cases Performance Using VPN

<table>
<thead>
<tr>
<th>Use Cases</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>startVM</td>
<td>2.5</td>
</tr>
<tr>
<td>stopVM</td>
<td>2</td>
</tr>
<tr>
<td>pauseVM</td>
<td>2</td>
</tr>
<tr>
<td>createVM</td>
<td>4.5</td>
</tr>
<tr>
<td>deleteVM</td>
<td>2.5</td>
</tr>
</tbody>
</table>

- **phone w. vpn**
- **pc w. vpn**
System Evaluation

SSH Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Time (milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ls</td>
<td>4</td>
</tr>
<tr>
<td>ps -ef</td>
<td>12</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
</tr>
</tbody>
</table>

(phone) vs (pc)
## System Evaluation

### Additional Performance Statistics

<table>
<thead>
<tr>
<th>Action</th>
<th>Average Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>startNewVM</td>
<td>1.5 min.</td>
</tr>
<tr>
<td>Transition between activities</td>
<td>0.0592 sec.</td>
</tr>
<tr>
<td>SSH commands</td>
<td>2 millisec.</td>
</tr>
</tbody>
</table>
Use Cases Demo
1. Login
2. Select a course from the list
3. Select assignment from the list
4. Tap Create Virtual Machine
Student View VM

Student is on the Courses view

1. Tap Virtual Machine button
2. Select the machine from the list
3. Tap the View button
4. Enter username
5. Enter password
Instructor Edit Settings

1. Login
2. Tap Virtual Machine Templates button
3. Tap on the Settings button.
4. Adjust machine’s settings
5. Save
Instructor is on the Courses view

1. Tap Virtual Machine button

2. Select machine to delete from list

3. Press Delete button

4. Confirm action
Performance Evaluation Demo
User is on VM Controls screen

1. Tap start VM

2. Press stop VM

3. Repeat 3 times

4. Open file ‘startVMDemo.txt ’ located on SD card
System SSH Commands

User is on VM Controls screen

1. Tap view VM
2. Enter username
3. Enter password
4. Type ‘date +%N;(any cmd) ;date +%N’
5. Subtract to last from first to get execution time.
Questions?