# Teaching GUI in Operating Systems Courses

Lior Shamir
Department of Computer Science, Michigan Tech
1400 Townsend Dr., Houghton, MI 49931

**Abstract**

Graphical User Interface (GUI) is an integral part of many modern operating systems, and GUI-based features have been attracting a considerable attention from operating system developers. However, while OS concepts such as CPU scheduling, memory management, disk management, file systems and security are typically covered in operating system courses, GUI is poorly studied in colleges in the sense of operating systems.
We describe an operating systems course section that discusses GUI in the sense of modern operating systems. Due to the increasing importance of GUI in some of the world's most popular operating systems, we propose to include a GUI section in the curriculum of undergraduate operating systems courses. The proposed section discusses the different GUI components used by operating systems, GUI resource allocation, window painting and handling and the programmer user interface.

## GUI Section Details

1. History of GUI Systems
- LISA, X-Windows, MS-Windows

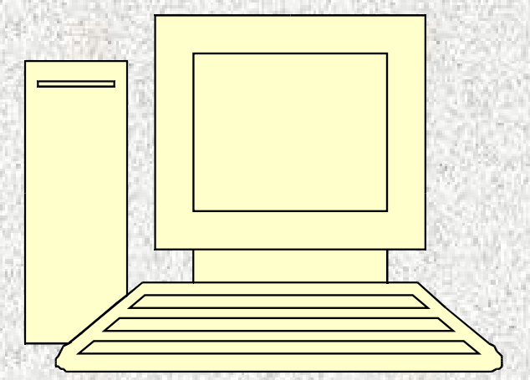
2. GDI and *User* Windows subsystems
- Purpose, scope and interfaces
- Advantages and disadvantages of GDI
- Solutions to the GDI performance problem for animation and video games (DirectX, OpenGL)

3. Window Handling
- Z-order
- Window painting
- Parent/Child windows
- Pre-defined control windows (buttons, textboxes, etc')
- Window messages
- Mouse and keyboard I/O

4. GUI Programmer Interface
- Window procedures
- Window message processing
- GUI API (*CreateWindow*, *FindWindow*, *SetWindowPos*, etc')
- Reading and setting window attributes

## The Computing Environment:

Hardware: PC



Operating System: Windows XP

Development Environment: Visual Studio 2005

- Powerful integral GUI
- Popularity
- An opportunity to give students experience in Windows programming
- Students considered working with Visual Studio as an important advantage
- Mastering Visual C++ is not beyond the capabilities of the average CS student, but a learning curve still exists. Therefore, detailed explanation (walkthrough) of how to start, compile and debug a project was provided.

## Programming Examples and Home Assignments

Creating windows – Creating user windows, creating control windows, registering window classes, writing window procedures.

Window positioning – Getting and setting the x, y and z coordinates of windows.

Window messages – Finding window handles and sending window messages.

Key logger – A simple program that records user key strokes and mouse clicks, and can later repeat the same sequence.
Students used the APIs *keybd_event*, *GetCursorPos* and *SetCursorPos*.

Find and close open browsers – A simple code that finds and closes all open browser windows.
Students used *FindWindow*, *GetNextWindow*, *GetTopWindow*, and *SendMessage*.

**Conclusion:**
Due to the increasing importance of GUI in modern operating systems, we propose to include a brief discussion of this topic in undergraduate operating systems courses. The proposed GUI section includes top-level GUI components and design, window painting and handling, GUI messages, and GUI programmer interface. Simple programming assignments were based on Visual Studio 2005 under MS-Windows XP.
The results of the student evaluation and the active discussions in class indicated that students were particularly interested in the subject, and believed that the topic was essential both to their knowledge as computer scientists and for practical real-life programming.

**Student Evaluation:**
An Anonymous survey was given late in the semester to the students enrolled in the operating systems course (CS-4411).
The questions and the survey results were the following:

In scale of 1 to 5, how do you think each of the following topics contributed to your general understanding of computer systems?

| Section | Average Student Evaluation (1 to 5) |
|---|---|
| Process scheduling | 4.0 |
| Process synchronization | 4.3 |
| Windows multi-threading | 4.4 |
| Windows GUI | 4.2 |
| Memory management | 4.5 |
| Inter-process communication | 4.1 |
| Deadlocks | 3.5 |
| Secondary storage devices | 3.7 |
| Interrupts | 3.9 |

Do you find the following topics practical for real-life programming?

| Section | Average Student Evaluation (1 to 5) |
|---|---|
| Process scheduling | 3.7 |
| Process synchronization | 4.1 |
| Windows multi-threading | 4.8 |
| Windows GUI | 4.8 |
| Memory management | 4.0 |
| Inter-process communication | 4.4 |
| Deadlocks | 2.5 |
| Secondary storage devices | 3.1 |
| Interrupts | 3.0 |

On which of the following topics would you like to learn more?

| Section | Average Student Evaluation (1 to 5) |
|---|---|
| Process scheduling | 3.5 |
| Process synchronization | 3.8 |
| Windows multi-threading | 4.3 |
| Windows GUI | 4.7 |
| Memory management | 4.1 |
| Inter-process communication | 3.8 |
| Deadlocks | 2.9 |
| Secondary storage devices | 2.6 |
| Interrupts | 2.6 |