

CI5250 – Computing Systems

Operating Systems Assignment (part of TB1)

Deadline: 2nd of December 2019 @ 23:59

Assignment Overview

The Windows Operating Systems family is the most widely used operating system for personal computers. This assignment is about the Windows operating system architecture and functionality. The assignment consists of three questions. Question one is concerned with the Windows operating system architecture, question two covers the message and event handling which is at the heart of an interactive system while question 3 is concerned with resource handling. You need to answer ALL questions. This is an individual piece of work. Please answer questions in full sentences. No bullet points. Put your answers into the provided answer sheet which should have your name and k-number at the top. You must reference any external sources that you have used to derive your answers .

Submit your answer sheet as a WORD or PDF document via CANVAS. There are **30 marks** in total. This assignment counts for 15% of the total mark of this module.

Question 1: Windows Architecture - (10 marks)

Describe the general architecture of the Windows operating system. Provide a system diagram and explain the major components of the system. Highlight two innovations in the Windows operating system design. Discuss how the architecture of the Windows system contributed to the success in the market.

Question 2: Windows Event Handling - (10 marks)

The Windows Operating System and applications are event-driven and retrieve messages from message queues. The messages are then further dispatched to the appropriate **windows procedures** for each open window in an application.

- a) Discuss how event-driven systems are beneficial for interactive applications.
- b) Take a look at the source code of a **WndProc** function example below. Analyse how the windows procedure (**WndProc**) is decoding a WM_COMMAND event message to display a Dialog Box. You also need to describe the role of the parameter variables. Refer to the source code in your description.

```

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    int wmId, wmEvent;
    PAINTSTRUCT ps;
    HDC hdc;

    switch (message)
    {
    case WM_COMMAND:
        wmId  = LOWORD(wParam);
        wmEvent = HIWORD(wParam);
        // Parse the menu selections:
        switch (wmId)
        {
        case IDM_ABOUT:
            DialogBox(hInst, MAKEINTRESOURCE(IDD_ABOUTBOX), hWnd, About);
            break;
        case IDM_EXIT:
            DestroyWindow(hWnd);
            break;
        default:
            return DefWindowProc(hWnd, message, wParam, lParam);
        }
        break;
    case WM_PAINT:
        hdc = BeginPaint(hWnd, &ps);
        // TODO: Add any drawing code here...
        EndPaint(hWnd, &ps);
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
    default:
        return DefWindowProc(hWnd, message, wParam, lParam);
    }
    return 0;
}

```

Question 3: Resource Management (10 marks)

The Operating System manages resources such as memory, files and CPU time which are assigned to processing entities such as processes and threads. Processes request resources from the operating system through appropriate system API function calls.

- a) Run the **ResourcesApp.exe** on the command line and analyse the source code as well. The application requests and releases processing and memory resources. Through which system function calls is this achieved?

- b) Run the application with increasing processing demands (1,2,...16) processing threads. The application will print out how long it took to complete the processing for all threads. Populate the table below with the timings and plot a line graph using Excel or any other spreadsheet software. Copy the graph and table into your answer sheet. Describe how the OS scheduler is distributing the processing between different CPU cores. This can be observed in the performance tab of the Task Manager or the Process Explorer from the sysinternals tools. How many CPU cores are busy? Give an explanation for your observations. How efficient is the computer system at handling the increasing processing demand?

Populate the table:

Processing threads	1	2	3	4	...	16
Time to complete (sec) all cores						