Systemy Operacyjne

Ćwiczenia, system Windows

Terminy zajęć

Zajęcia ćwiczeniowe z części dotyczącej systemu Windows odbywają się w sali D3.

a) Grupa C1

Data	13.10	27.10	10.11	24.11	8.12	22.12	12.01	26.01
Sala		D3		D3		D3		D3

b) Grupa C2

Data	6.10	20.10	3.11	17.11	1.12	15.12	5.01
Sala		D3		D3		D3	

c) Grupa C3

Data	13.10	27.10	10.11	24.11	8.12	22.12	12.01	26.01
Sala	D3		D3		D3		D3	

d) Grupa C4

Data	6.10	20.10	3.11	17.11	1.12	15.12	5.01
Sala	D3		D3		D3		D3

System punktowy

Student samodzielnie, korzystając z wiadomości zdobytych na laboratorium, wykładzie oraz dokumentacji MFC i innych źródeł, wykonuje polecenia zapisane w punktach poniżej. Poprawna realizacja każdego podpunktu jest nagradzana plusem.

Wykonując każde ćwiczenie można uzyskać trzy punkty. Po wykonaniu każdego podpunktu, student powinien zgłosić się i poczekać aż prowadzący zajęcia zweryfikuje poprawność wykonania. Podpunkty należy wykonywać w kolejności.

Ćwiczenia w trybie stacjonarnym należy realizować **wyłącznie** na maszynie wirtualnej SO_BD_IO. Podczas zajęć należy samodzielnie projektować interfejs użytkownika. **Czytelność i praktyczność interfejsu podlega ocenie**. Sytuacje sporne rozstrzyga prowadzący.

Praca zdalna

Podczas pracy zdalnej należy realizować zadania samodzielnie. Instrukcja pobrania maszyny wirtualnej SO_BD_IO znajduje się na stronie prowadzącego. Zadania można realizować również na dowolnym innym systemie z rodziny Windows, warto jednak mieć na uwadze, że może to prowadzić do problemów, które nie występują na maszynie SO_BD_IO

(przede wszystkim: problemów z kodowaniem znaków w standardzie UNICODE, zamiast ASCII).

Zadania realizowane zdalnie będą oceniane wyłącznie wtedy, gdy zostaną wysłane prowadzącemu **w trakcie zajęć**. Wykrycie przez prowadzącego plagiatu będzie oznaczać otrzymanie trzech minusów i anulowanie jakichkolwiek plusów, które były otrzymane zdalnie. Zadania należy wysłać zbiorczo, po zakończeniu wszystkich podpunktów. Wystarczające jest wysłanie jednej wiadomości zawierającej archiwum z projektem.

Ćwiczenie 1: wprowadzenie do środowiska MFC

a) Opracuj w środowisku MFC aplikację „kalkulator". Aplikacja powinna mieć dwa pola edycyjne, w których można wpisać liczby i przycisk „+" po naciśnięciu, którego program obliczy wynik dodawania liczb. Ostatnim elementem programu powinno być pole edycyjne/pole tekstowe, w którym program wyświetli obliczony wynik.

b) Rozbuduj program „kalkulator" poprzez zamianę przycisku dodawania na rozwijaną listę opcji oraz dodanie przycisku „=". Po naciśnięciu przycisku „=", program powinien obliczyć wynik działania. Lista powinna zawierać cztery operacje: dodawanie, odejmowanie, mnożenie i dzielenie. Operacje wykonuj na liczbach zmiennoprzecinkowych.

c) Zmień program z podpunktu 1b., w taki sposób, aby zawierał tylko jedno pole edycyjne, przycisk „=" i pole wyniku. Po wpisaniu w pole edycyjne operacji w formacie „[argument1] [operacja] [argument2]" (np. „5 * -6.5") i naciśnięciu „=", program powinien obliczyć wynik działania.

Przydatne frazy i metody: GetWindowText, SetWindowText, sprintf, sscanf, atoi, atof, itoa, ftoa, CString to int, int to CString, CString.Format

Przydatne materiały:
- **dokumentacja MFC:** https://docs.microsoft.com/pl-pl/cpp/mfc/reference/mfc-classes?view=vs-2015
- **Tworzenie projektu (początek na przedostatniej stronie):** http://materialy.prz-rzeszow.pl/pracownik/pliki/42/so2.pdf

---

Ćwiczenie 2: potoki nazwane

a) Uruchom w tle program „npserwer32", który znajduje się w materiałach dodatkowych do laboratorium 4 (http://materialy.prz-rzeszow.pl/pracownik/pliki/28/so_procesy.zip). Jest to serwer potoku nazwanego *\\.\pipe\sysopnp*. Opracuj w środowisku MFC aplikację, która zawiera przycisk „Wyślij wiadomość". Wciśnięcie przycisku powinno wysłać do potoku

komunikat „PUT testowa wiadomosc”. Odpowiedź serwera wyświetl w wyskakującym okienku, dodatkowym polu edycyjnym lub zaprezentuj w inny, dowolny, ale widoczny sposób.

b) Rozbuduj program z podpunktu 2a., w taki sposób, aby zawierał przycisk „Pobierz” i pole edycyjne z opcją „tylko do odczytu”. Po wciśnięciu „Pobierz”, program powinien wysłać do potoku *\\.\pipe\sysopnp* komunikat „GET”. Zapisz otrzymany wynik do pola edycyjnego. Wynikiem komendy GET powinna być lista wszystkich wiadomości wysłanych do potoku.

c) Rozbuduj program z podpunktu 2b. o dodatkowe pole edycyjne. Zmień działanie przycisku „Wyślij wiadomość”, aby zamiast wysyłać „PUT testowa wiadomosc", wysyłał „PUT „, wraz z zawartością nowoutworzonego pola edycyjnego. Pole z odebranymi wiadomościami powinno wyświetlać każdą wiadomość **w osobnej linii**.

Przydatne frazy i metody: GetWindowText, SetWindowText, CallNamedPipe, CString.Format,

Przydatne materiały:

- dokumentacja MFC: https://docs.microsoft.com/pl-pl/cpp/mfc/reference/mfc-classes?view=vs-2015
- Praca z serwerem potoku: http://materialy.prz-rzeszow.pl/pracownik/pliki/42/so4.pdf

---

Ćwiczenie 3: skrzynki pocztowe

a) Utwórz w środowisku MFC program, który poprawnie tworzy nową skrzynkę pocztową, o dowolnej, poprawnej, unikalnej, wybranej przez Ciebie nazwie. Program powinien zawierać przycisk „Sprawdź” i pole edycyjne/tekstowe (tylko do odczytu). Po wciśnięciu „Sprawdź”, w polu tym powinna się wyświetlać ilość wszystkich wiadomości zapisanych w skrzynce pocztowej od momentu jej utworzenia. Zaimplementuj mechanizm testujący poprawność *utworzenia* skrzynki.

b) Rozbuduj program z podpunktu 3a. Dodaj przycisk „Wyślij” i kolejne pole edycyjne. Wciśnięcie przycisku „Wyślij” powinno wysłać do skrzynki wiadomość z wpisanym w polu edycyjnym tekstem.

c) Rozbuduj program z podpunktu 3b. Dodaj pole edycyjne (tylko do odczytu) oraz przycisk „Odczytaj”. Po wciśnięciu przycisku, pobierz treść **wszystkich** wiadomości jakie zostały kiedykolwiek wysłane do skrzynki i wyświetl je w nowoutworzonym polu tekstowym. Pole z odebranymi wiadomościami powinno wyświetlać każdą wiadomość **w osobnej linii**.

Przydatne frazy i metody: GetWindowText, SetWindowText, CreateMailslot, GetMailslotInfo, CFile, CFile.Read, CFile.Write, CString.Format,

Przydatne materiały:
- **dokumentacja MFC:** https://docs.microsoft.com/pl-pl/cpp/mfc/reference/mfc-classes?view=vs-2015

Ćwiczenie 4: producent i konsument

a) Utwórz w środowisku MFC program, który tworzy dwa wątki robocze. Zdefiniuj 10-elementową tablicę typu logicznego i zainicjuj samymi wartościami „false”. Pierwszy wątek (zwany „producentem”) powinien co 1.5s ustawiać wartość kolejnego elementu tablicy na „true”. W razie zapełnienia bufora wartościami „true”, wątek powinien poczekać, aż w buforze będą jakieś miejsca i wznowić swoje działanie. Drugi wątek (zwany „konsumentem”) powinien co 2s sprawdzać, czy jest dostępny element z wartością „true” i zmieniać jego wartość na „false”. Umieść w programie pole edycyjne (tylko do odczytu) które będzie zawierać informacje o tym, ile elementów tablicy ma wartość „true”.

b) Rozbuduj pogram z punktu 4a. Zmień interfejs w taki sposób, aby dokładnie pokazywał, które elementy tablicy mają wartość „false”, a które „true”. Automatycznie odświeżaj interfejs co jedną dziesiątą sekundy.

c) Rozbuduj program z punktu 4b. Dodaj dwa pola edycyjne, które pozwolą ustalić czas pracy producenta i konsumenta.