|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | |
| Институт информационных технологий (ИТ) | |
| Кафедра инструментального и прикладного программного обеспечения (ИиППО) | |

|  |  |
| --- | --- |
| **ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №9** | |
| **по дисциплине** | |
| «Системное программное обеспечение»  по теме: «Межпроцессное взаимодействие» | |
|  | |
| Выполнил студент группы ИКБО-16-18 | Шамян Г.Г. |
| Принял преподаватель | Волков М.Ю. |

|  |  |  |
| --- | --- | --- |
| Работа выполнена | «\_\_»\_\_\_\_\_\_\_2021 г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2021 г. | *(подпись руководителя)* |

Москва 2021

**Цель работы**

Изучение механизмов межпроцессного взаимодействия (InterProcess Communication) в Windows NT; получение практических навыков по использованию Win32 API для программирования механизмов IPC

**Задание**

1. Написать программу с использованием одного из способов межпроцессного взаимодействия.

**Ход работы**

Итоговая программа представлена в листинге 1. Результат работы представлен на рисунках 1-2.

Листинг 1

**Main.cpp**

#include <windows.h>

#include <tchar.h>

#include <stdio.h>

#include <strsafe.h>

HANDLE hSlot;

LPCTSTR SlotName = TEXT("\\\\.\\mailslot\\sample\_mailslot");

BOOL WINAPI MakeSlot(LPCTSTR lpszSlotName)

{

hSlot = CreateMailslot(lpszSlotName,

0, // no maximum message size

MAILSLOT\_WAIT\_FOREVER, // no time-out for operations

(LPSECURITY\_ATTRIBUTES)NULL); // default security

if (hSlot == INVALID\_HANDLE\_VALUE)

{

printf("CreateMailslot failed with %d\n", GetLastError());

return FALSE;

}

return TRUE;

}

BOOL WriteSlot(HANDLE hSlot, LPCTSTR lpszMessage)

{

BOOL fResult;

DWORD cbWritten;

fResult = WriteFile(hSlot,

lpszMessage,

(DWORD)(lstrlen(lpszMessage) + 1) \* sizeof(TCHAR),

&cbWritten,

(LPOVERLAPPED)NULL);

if (!fResult)

{

printf("WriteFile failed with %d.\n", GetLastError());

return FALSE;

}

printf("Slot written to successfully.\n");

return TRUE;

}

void main()

{

MakeSlot(SlotName);

HANDLE hFile;

hFile = CreateFile(SlotName,

GENERIC\_WRITE,

FILE\_SHARE\_READ,

(LPSECURITY\_ATTRIBUTES)NULL,

OPEN\_EXISTING,

FILE\_ATTRIBUTE\_NORMAL,

(HANDLE)NULL);

if (hFile == INVALID\_HANDLE\_VALUE)

{

printf("CreateFile failed with %d.\n", GetLastError());

}

while (TRUE)

{

WriteSlot(hFile, TEXT("Valhalla calling you"));

Sleep(3000);

}

}

**Client.cpp**

#include <windows.h>

#include <tchar.h>

#include <stdio.h>

#include <strsafe.h>

HANDLE hSlot;

LPCTSTR SlotName = TEXT("\\\\.\\mailslot\\sample\_mailslot");

BOOL ReadSlot()

{

DWORD cbMessage, cMessage, cbRead;

BOOL fResult;

LPTSTR lpszBuffer;

TCHAR achID[80];

DWORD cAllMessages;

HANDLE hEvent;

OVERLAPPED ov;

cbMessage = cMessage = cbRead = 0;

hEvent = CreateEvent(NULL, FALSE, FALSE, TEXT("ExampleSlot"));

if (NULL == hEvent)

return FALSE;

ov.Offset = 0;

ov.OffsetHigh = 0;

ov.hEvent = hEvent;

fResult = GetMailslotInfo(hSlot, // mailslot handle

(LPDWORD)NULL, // no maximum message size

&cbMessage, // size of next message

&cMessage, // number of messages

(LPDWORD)NULL); // no read time-out

if (!fResult)

{

printf("GetMailslotInfo failed with %d.\n", GetLastError());

return FALSE;

}

if (cbMessage == MAILSLOT\_NO\_MESSAGE)

{

printf("Waiting for a message...\n");

return TRUE;

}

cAllMessages = cMessage;

while (cMessage != 0) // retrieve all messages

{

// Create a message-number string.

StringCchPrintf((LPTSTR)achID,

80,

TEXT("\nMessage #%d of %d\n"),

cAllMessages - cMessage + 1,

cAllMessages);

// Allocate memory for the message.

lpszBuffer = (LPTSTR)GlobalAlloc(GPTR,

lstrlen((LPTSTR)achID) \* sizeof(TCHAR) + cbMessage);

if (NULL == lpszBuffer)

return FALSE;

lpszBuffer[0] = '\0';

fResult = ReadFile(hSlot,

lpszBuffer,

cbMessage,

&cbRead,

&ov);

if (!fResult)

{

printf("ReadFile failed with %d.\n", GetLastError());

GlobalFree((HGLOBAL)lpszBuffer);

return FALSE;

}

// Concatenate the message and the message-number string.

StringCbCat(lpszBuffer,

lstrlen((LPTSTR)achID) \* sizeof(TCHAR) + cbMessage,

(LPTSTR)achID);

// Display the message.

\_tprintf(TEXT("Contents of the mailslot: %s\n"), lpszBuffer);

GlobalFree((HGLOBAL)lpszBuffer);

fResult = GetMailslotInfo(hSlot, // mailslot handle

(LPDWORD)NULL, // no maximum message size

&cbMessage, // size of next message

&cMessage, // number of messages

(LPDWORD)NULL); // no read time-out

if (!fResult)

{

printf("GetMailslotInfo failed (%d)\n", GetLastError());

return FALSE;

}

}

CloseHandle(hEvent);

return TRUE;

}

BOOL WINAPI MakeSlot(LPCTSTR lpszSlotName)

{

hSlot = CreateMailslot(lpszSlotName,

0, // no maximum message size

MAILSLOT\_WAIT\_FOREVER, // no time-out for operations

(LPSECURITY\_ATTRIBUTES)NULL); // default security

if (hSlot == INVALID\_HANDLE\_VALUE)

{

printf("CreateMailslot failed with %d\n", GetLastError());

return FALSE;

}

return TRUE;

}

void main()

{

MakeSlot(SlotName);

while (TRUE)

{

ReadSlot();

Sleep(3000);

}

}

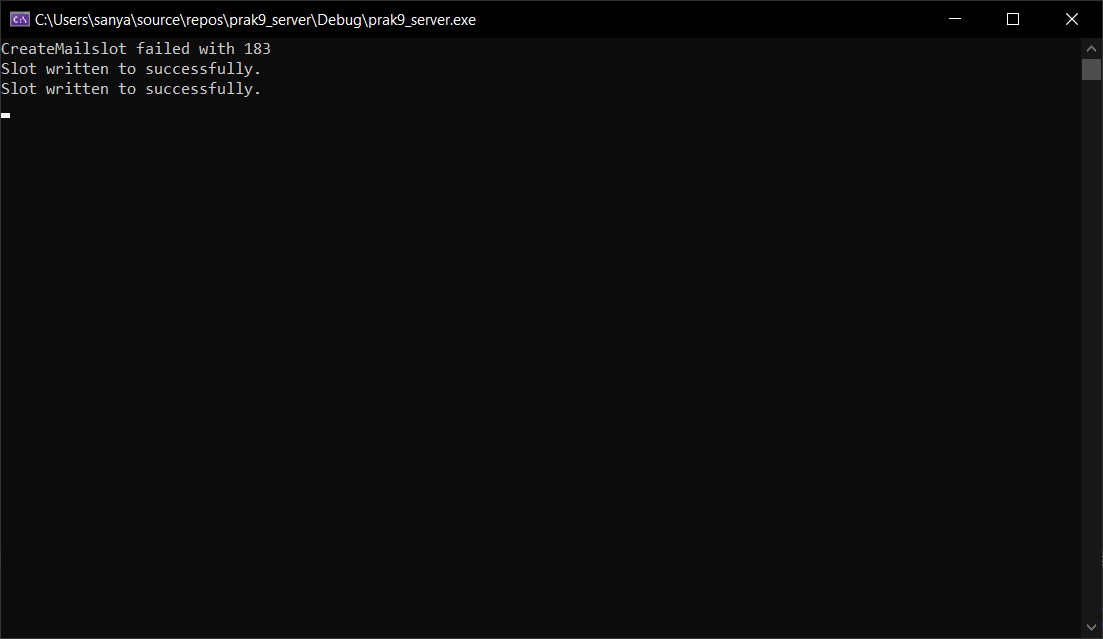


Рис. 1 – Создание mailslot

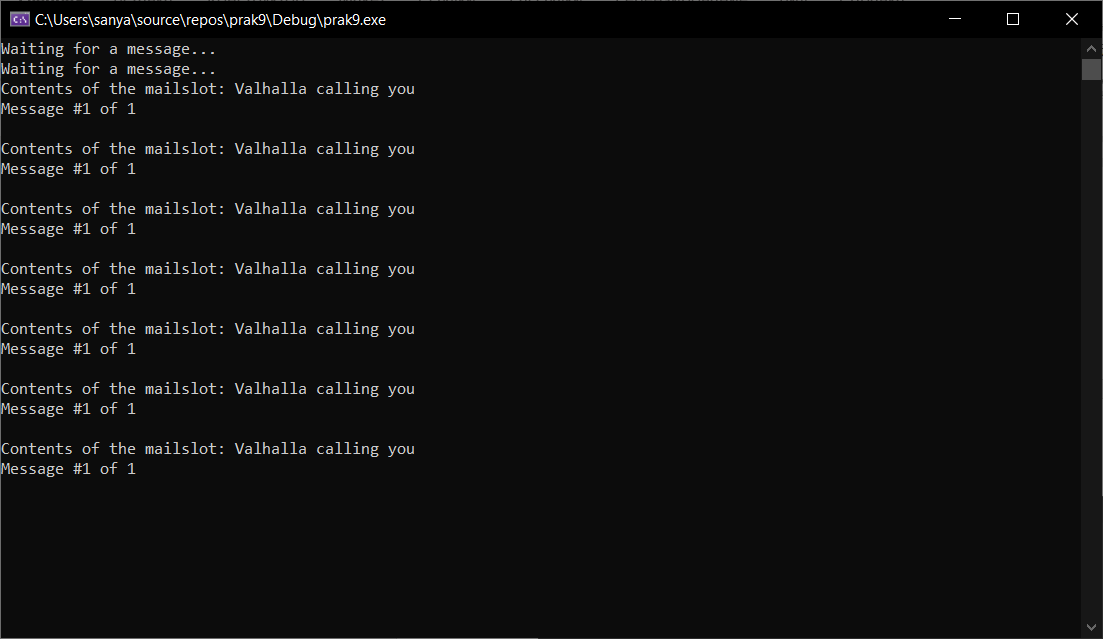


Рис. 2 – Чтение mailslot

**Вывод**

В ходе практической работы мы изучили механизмы межпроцессорного взаимодействия.

**Список использованных источников**

1. Пошаговое руководство. Создание классического оконного приложение [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/cpp/windows/walkthrough-creating-windows-desktop-applications-cpp?view=msvc-160
2. Администрирование в Windows 10. [Электронный ресурс] : [https://docs.microsoft.com/ru-ru/windows/client-management/administrative-tools-in-windows-10 /](https://docs.microsoft.com/ru-ru/windows/client-management/administrative-tools-in-windows-10%20/) Ричард Ворд 2020 — 100 с.
3. «Тайм-менеджмент для системных администраторов». / Лимончелли Т.А. М.: Символ-плюс 2007 — 247 с. ISBN:978-5-93286-090-8