**БУ ВО Ханты-Мансийского автономного округа – Югры**

**«Сургутский государственный университет»**

**Политехнический институт**

**Кафедра автоматики и компьютерных систем**

**ОТЧЁТ**

**по лабораторной работе №4**

**по дисциплине: «Операционные системы»**

**Выполнил: студент группы №609-21**

**Шумилов И. Д.**

**Принял: старший преподаватель кафедры АиКС**

**Кривицкая М. А.**

**Сургут**

**2024г.**

Цель работы: изучить принципы взаимодействия процессов (потоков), функции, структуры данных и объекты ядра, использующиеся для организации этого взаимодействия и получить навыки практического использования прикладного программного интерфейса.

**Задание на лабораторную работу:**

Разработать и отладить прогрпмму, выполняющую следующие функции:

1. Разработать и отладить программу, представляющую процесс-родитель и выполняющую следующие функции:
   1. создание анонимного канала;
   2. создание (запуск) процесса-потомка (см. п.2 ип.3), у которого стандартный поток ввода перенаправлен на поток ввода анонимного канала;
   3. передачу потока данных через канал процессу-потомку.
2. Разработать и отладить программу, представляющую процесс-потомок и выполняющую следующую функции:
   1. получение данных от процесса-родителя через стандартный поток ввода;
   2. форматированный вывод полученных данных в стандартный поток вывода;
3. Проверить работу канала передачи данных при организации взаимодействия между процессом-родителем и процессом-потомком для двух вариантов:
   1. в качестве процесса-потомка использовать разработанную программу;
   2. в качестве процесса-потомка использовать стандартную утилиту Windows.

Листинг 1. Код программы родителя на С++:

* 1. #include <Windows.h>
  2. #include <iostream>
  4. #define BUFSIZE 4096
  6. int main(int argc, char\*\* argv) {
  7. HANDLE hChildWr = NULL;
  8. HANDLE hChildRd = NULL;
  10. SECURITY\_ATTRIBUTES sa;
  11. sa.nLength = sizeof(SECURITY\_ATTRIBUTES);
  12. sa.bInheritHandle = TRUE;
  13. sa.lpSecurityDescriptor = NULL;
  15. if (!CreatePipe(&hChildRd, &hChildWr, &sa, NULL))
  16. std::cout << "Failed creating pipe" << std::endl;
  18. STARTUPINFO si;
  19. PROCESS\_INFORMATION pi;
  21. ZeroMemory(&si, sizeof(si));
  22. ZeroMemory(&pi, sizeof(pi));
  24. si.cb = sizeof(STARTUPINFO);
  25. si.hStdInput = hChildRd;
  26. si.dwFlags |= STARTF\_USESTDHANDLES;
  28. std::wstring CommandLine(L"C:\\Users\\qvarkk\\Desktop\\laba4 po os fix\\Child\\x64\\Debug\\Child.exe");
  29. LPWSTR lpwCmdlLine = &CommandLine[0];
  31. if (!CreateProcess(NULL, lpwCmdlLine, NULL, NULL, TRUE, CREATE\_NEW\_CONSOLE, NULL, NULL, &si, &pi))
  32. std::cout << "Failed creating process" << std::endl;
  34. DWORD dwWritten;
  35. CHAR wrBuf[BUFSIZE] = "this string has 30 characters.";
  37. if (!WriteFile(hChildWr, wrBuf, (DWORD)sizeof(wrBuf), &dwWritten, NULL))
  38. std::cout << "Failed writing to pipe" << std::endl;
  40. if (!CloseHandle(hChildWr))
  41. std::cout << "Failed closing stdin wr handle" << std::endl;
  43. CloseHandle(pi.hProcess);
  44. CloseHandle(pi.hThread);
  46. return 0;
  47. }

**Пояснение к программе-родителю (листинг 1):**

Создание анонимного канала происходит посредством функции CreatePipe() на строке 16. В качестве параметров ей были переданы указатели на ранее созданные в строках 7–8 дескрипторы, в которые и были записаны дескрипторы записи и чтения анонимного канала. В структуре SECURITY\_ATTRIBUTES для данной функции были указаны поля bInheritHandle как TRUE и lpSecurityDescriptor как NULL. Затем на строке 32 был создан процесс программы-потомка Child.exe с перенаправленным дескриптором потока ввода. После чего на строке 38 данные из строки wrBuf были записаны в анонимный канал.

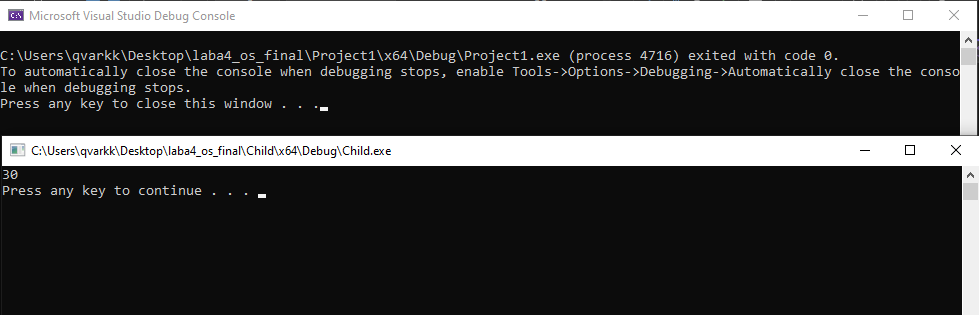
Листинг 2. Код программы предка на С++:

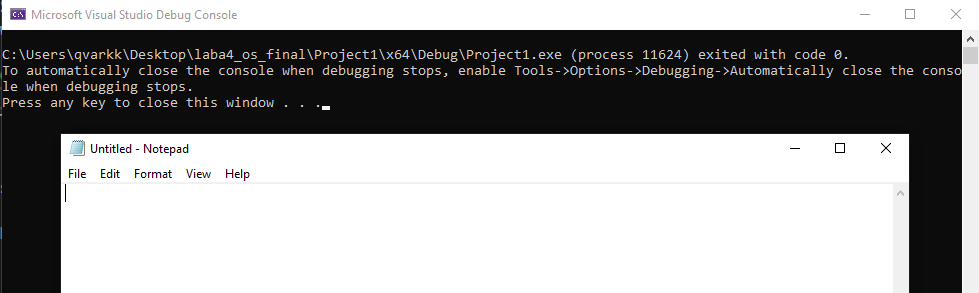
* 1. #include <Windows.h>
  2. #include <iostream>
  4. #define BUFSIZE 4096
  6. int main(int argc, char\*\* argv) {
  7. HANDLE hStdin = GetStdHandle(STD\_INPUT\_HANDLE);
  9. CHAR buffer[BUFSIZE];
  10. DWORD dwRead;
  11. BOOL bSuccess = FALSE;
  13. if (!ReadFile(hStdin, buffer, BUFSIZE, &dwRead, NULL))
  14. return -1;
  15. int i;
  16. for (i = 0; buffer[i] != '\0'; i++);
  17. std::cout << i << std::endl;
  19. system("pause");
  21. return 0;
  22. }

**Пояснение к программе-потомку (листинг 2):**

Сперва на строке 7 происходит получение стандартного дескриптора ввода, после чего функцией ReadFile() содержимое потока ввода записывается в переменную buffer, после чего происходит подсчет символов buffer и вывод этого числа.

На рис. 1 приведен пример работы программы-родителя.

Рис. 1 - Пример работы программы-родителя с программой-потомком

Рис. 2 - Пример работы программы-родителя с утилитой Windows notepad.exe

**Вывод:** былиизучены принципы взаимодействия процессов (потоков), функции, структуры данных и объекты ядра, использующиеся для организации этого взаимодействия, и были получены навыки практического использования прикладного программного интерфейса.