

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

**Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»**

**Лабораторная работа №3
по курсу «Операционные системы»**

**Выполнил: М. С. Разананирина
Группа: М8О-208БВ-24
Преподаватель: Е. С. Миронов**

Москва, 2025

Условие

Родительский процесс создает дочерний процесс. Первой строкой пользователь в консоль родительского процесса вводит имя файла, которое будет использовано для открытия File с таким именем на запись. Родительский и дочерний процесс должны быть представлены разными программами. Родительский процесс принимает от пользователя строки произвольной длины и пересылает их через memory-mapped files. Процесс child проверяет строки на валидность правилу. Если строка соответствует правилу, то она выводится в файл, иначе через memory-mapped files выводится информация об ошибке. Родительский процесс полученные от child ошибки выводит в стандартный поток вывода.

Цель работы: Приобретение практических навыков в:

- Освоение принципов работы с файловыми системами
- Обеспечение обмена данных между процессами посредством технологии «File mapping»

Задание: Разработать программу, состоящую из двух процессов — родительского и дочернего, взаимодействующих через отображаемые файлы (memory-mapped files).

Родительский процесс должен:

- Запрашивать у пользователя имя файла и передавать его дочернему процессу через командную строку;
- Принимать от пользователя строки с числами и передавать их дочернему процессу через разделяемую память;
- Ожидать обработки данных дочерним процессом через механизм синхронизации в разделяемой памяти;
- Завершать работу при получении сигнала об ошибке деления на ноль через разделяемую память.

Дочерний процесс должен:

- Получить от родительского процесса имя файла через аргументы командной строки и открыть его для записи;
- Принимать строки с числами от родительского процесса через разделяемую память;
- Выполнять операцию последовательного деления первого числа на последующие (кумулятивное деление);
- Записывать результаты вычислений в файл;
- Проверять деление на ноль и при обнаружении устанавливать флаг ошибки в разделяемой памяти;
- Завершать работу при обнаружении деления на ноль с аварийным кодом возврата.

Вариант: 3

Метод решения

Для решения задачи применена архитектура с двумя процессами (родительским и дочерним), взаимодействующими через технологию File Mapping (отображаемые файлы). Родительский процесс отвечает за взаимодействие с пользователем и координацию работы, тогда как дочерний процесс выполняет вычислительную логику и работу с файлами.

Основной алгоритм работы

1. **Инициализация:** Родительский процесс создает разделяемую память (shared memory) и запрашивает у пользователя имя файла для сохранения результатов
2. **Запуск процесса:** Создание дочернего процесса с передачей имени файла через аргументы командной строки
3. **Синхронизация процессов:** Оба процесса подключаются к общей области памяти для обмена данными
4. **Обработка данных:**
 - Родительский процесс читает числовые данные от пользователя и записывает их в разделяемую память
 - Дочерний процесс отслеживает флаг `data_ready`, читает числа из разделяемой памяти
 - Дочерний процесс выполняет кумулятивное деление первого числа на последующие с проверкой на деление на ноль
 - Результаты вычислений записываются в файл, устанавливается флаг `processing_complete`
5. **Завершение работы:** Корректное закрытие процессов при получении команды `exit` или ошибке деления на ноль

Особенности реализации

Для обеспечения межпроцессного взаимодействия использована технология File Mapping Windows API. Реализована синхронизация процессов через флаги в разделяемой памяти. Обработка ошибок включает проверку системных вызовов, валидацию входных данных и аварийное завершение при делении на ноль.

Описание программы

Программа реализована в модульном стиле и состоит из четырех основных компонентов.

Модуль `parent.c`

Реализует логику родительского процесса:

- Создание и управление дочерним процессом через функции библиотеки
- Создание и инициализация разделяемой памяти для межпроцессного взаимодействия

- Взаимодействие с пользователем (ввод имени файла и числовых данных)
- Запись данных в разделяемую память и ожидание обработки дочерним процессом
- Обработка флага `division_by_zero` для аварийного завершения работы
- Координация корректного завершения работы процессов

Модуль `child.c`

Содержит вычислительную логику и работу с файлами:

- Получение имени файла через аргументы командной строки
- Подключение к существующей разделяемой памяти
- Отслеживание флага `data_ready` для получения новых данных
- Парсинг числовых данных и валидация входных значений
- Выполнение кумулятивного деления чисел с проверкой на деление на ноль
- Запись промежуточных и финальных результатов в файл
- Установка флагов состояния в разделяемой памяти

Модуль `library.h/c`

Предоставляет кроссплатформенные абстракции и утилиты:

- **Структуры данных:** `process_t` (для управления процессами)
- **Функции управления процессами:** `CpProcessCreate`, `CpProcessWrite`, `CpProcessRead`, `CpProcessClose`
- **Функции работы с разделяемой памятью:** `CpCreateSharedMemory`, `CpOpenSharedMemory`, `CpCloseSharedMemory`
- **Вспомогательные функции:** `TrimNewline` - удаление символов новой строки
- **Кроссплатформенные обертки:** унифицированный API для Windows и Unix систем

Модуль `shared_data.h`

Определяет структуру данных для разделяемой памяти:

- `filename` - имя файла для записи результатов
- `numbers` - массив чисел для обработки
- `numbers_count` - количество чисел в массиве

- `data_ready` - флаг готовности данных для обработки
- `processing_complete` - флаг завершения обработки
- `division_by_zero` - флаг ошибки деления на ноль
- `terminate_process` - флаг запроса на завершение работы

Используемые системные вызовы

- **Windows API:** `CreateFileMapping`, `MapViewOfFile`, `OpenFileMapping`, `UnmapViewOfFile`
- **Управление процессами:** `CreateProcess`, `CreatePipe`, `ReadFile`, `WriteFile`, `CloseHandle`, `WaitForSingleObject`
- **Файловые операции:** `fopen`, `fclose`, `fprintf`, `fflush`
- **Обработка строк:** `strtok`, `strncpy`, `atoi`, `strlen`

Архитектура программы обеспечивает четкое разделение ответственности между процессами и использует современные механизмы межпроцессного взаимодействия через технологию File Mapping.

Результаты

В результате работы была разработана программа для межпроцессного взаимодействия с использованием технологии File Mapping, успешно функционирующая в операционной системе Windows и Unix-системах.

Ключевые особенности реализации

- **Межпроцессное взаимодействие через File Mapping:** Использование разделяемой памяти для обмена данными между родительским и дочерним процессами
- **Синхронизация процессов:** Координация работы процессов через флаги в разделяемой памяти (`data_ready`, `processing_complete`)
- **Надежная обработка данных:** Парсинг числовых входных данных с валидацией и контролем ошибок
- **Арифметическая безопасность:** Обнаружение и обработка деления на ноль с аварийным завершением обоих процессов
- **Корректное управление ресурсами:** Правильное закрытие процессов, файлов и дескрипторов разделяемой памяти при завершении

Пример работы программы

```
PS D:\MAI-year2\os\os_lab3\build>./parent.exe
Enter file name: test1
Child process created. Using FILE MAPPING for IPC.
Enter numbers separated by spaces (cumulative division):
For exit write 'exit' or empty line
>400 60 9
Waiting for child to process...
Processing complete. Results written to file.
>500 7 0
Waiting for child to process...
EMERGENCY: Division by zero detected! Shutting down...
Shutting down...
Parent process finished. Child exit code: 2
```

Содержимое файла test1:

```
Child process started. Output file: test1
Using FILE MAPPING for IPC with parent
Input: 400 60 9
Division results:
400.00 / 60.00 = 6.67
6.67 / 9.00 = 0.74
Final result: 0.74
---
Input: 500 7 0
ERROR: Division by zero! 500 / 0
EMERGENCY SHUTDOWN: Division by zero detected
```

Производительность

Программа демонстрирует стабильную работу при обработке числовых данных различного объема. Использование технологии File Mapping обеспечивает высокоскоростной обмен данными между процессами. Время отклика системы на ввод пользователя практически не отличается от времени работы обычных консольных приложений, что подтверждает эффективность выбранного подхода к организации межпроцессного взаимодействия.

Выводы

В ходе выполнения лабораторной работы были успешно достигнуты поставленные цели:

1. **Освоены принципы работы с файловыми системами:** На практике применены операции создания, открытия и записи в файлы. Реализована работа с файловой системой через стандартные функции языка Си (`fopen()`, `fprintf()`, `fclose()`) для сохранения результатов вычислений
2. **Обеспечен обмен данных между процессами посредством технологии File Mapping:** Организовано межпроцессное взаимодействие через разделяемую память с использованием Windows API (`CreateFileMapping()`, `MapViewOfFile()`). Реализована синхронизация процессов через флаги в общей структуре данных

3. **Создана программа для работы с процессами:** Разработана архитектура с родительским и дочерним процессами, где родительский процесс создает дочерний и координирует его работу через механизмы разделяемой памяти

4. **Реализована обработка системных ошибок:**

- Обработка ошибок создания и отображения разделяемой памяти
- Контроль ошибок открытия и записи в файлы
- Обнаружение и обработка арифметической ошибки деления на ноль
- Корректное освобождение ресурсов при аварийном завершении

Работа продемонстрировала эффективность использования технологии File Mapping для высокоскоростного обмена данными между процессами. Полученный опыт может быть применен при разработке сложных многопроцессных приложений, требующих интенсивного обмена данными.

Исходная программа

```
1 | #ifndef SHARED_DATA_H
2 | #define SHARED_DATA_H
3 |
4 | #define MAX_NUMBERS 100
5 | #define FILENAME_LENGTH 256
6 |
7 | typedef struct {
8 |     char filename[FILENAME_LENGTH];
9 |     int numbers[MAX_NUMBERS];
10 |    int numbers_count;
11 |    int data_ready;
12 |    int processing_complete;
13 |    int division_by_zero;
14 |    int terminate_process;
15 | } SharedData;
16 |
17 | #endif
```

Листинг 1: *shared_data.h*

```
1 | #ifndef LIBRARY_H
2 | #define LIBRARY_H
3 |
4 | #include <stddef.h>
5 | #include <stdio.h>
6 | #include <string.h>
7 |
8 | void TrimNewline(char* str);
9 |
10 | #ifdef _WIN32
11 | #include <windows.h>
12 | #else
13 | #include <sys/types.h>
14 | #endif
15 |
16 | #ifdef _WIN32
17 | typedef struct {
18 |     HANDLE handle;
19 |     HANDLE stdinWrite;
20 |     HANDLE stdoutRead;
21 | } process_t;
22 | #else
23 | typedef struct {
24 |     pid_t pid;
25 |     int stdin_fd;
26 |     int stdout_fd;
27 | } process_t;
28 | #endif
29 |
30 | int CpProcessCreate(process_t* proc, const char* path);
31 | int CpProcessWrite(process_t* proc, const char* data, size_t size);
32 | int CpProcessRead(process_t* proc, char* buffer, size_t size);
33 | int CpProcessClose(process_t* proc);
34 |
35 | size_t CpStringLength(const char* str);
```



```

36 | int CpStringContains(const char* str, const char* substr);
37 |
38 | void* CpCreateSharedMemory(const char* name, size_t size);
39 | void* CpOpenSharedMemory(const char* name, size_t size);
40 | void CpCloseSharedMemory(const char* name, void* ptr, size_t size);
41 |
42 | static inline const char* CpGetChildProcessName(const char* baseName) {
43 |     (void)baseName;
44 | #ifdef _WIN32
45 |     return "child.exe";
46 | #else
47 |     return "./child";
48 | #endif
49 | }
50 |
51 | #define CpWriteStdout(data, size) ((int)fwrite((data), 1, (size), stdout))
52 | #define CpWriteStderr(data, size) ((int)fwrite((data), 1, (size), stderr))
53 |
54 | #endif

```

Листинг 2: *library.h*

```

1 | #include "library.h"
2 | #include <stdlib.h>
3 |
4 | void TrimNewline(char* str) {
5 |     if (!str) return;
6 |     size_t len = strlen(str);
7 |     if (len > 0 && str[len - 1] == '\n') str[len - 1] = '\0';
8 | }
9 |
10 | #ifdef _WIN32
11 | #include <windows.h>
12 |
13 | void* CpCreateSharedMemory(const char* name, size_t size) {
14 |     HANDLE hMapFile = CreateFileMapping(
15 |         INVALID_HANDLE_VALUE,
16 |         NULL,
17 |         PAGE_READWRITE,
18 |         0,
19 |         (DWORD)size,
20 |         name);
21 |
22 |     if (hMapFile == NULL) {
23 |         fprintf(stderr, "Error creating shared memory (%lu)\n", GetLastError());
24 |         return NULL;
25 |     }
26 |
27 |     void* ptr = MapViewOfFile(
28 |         hMapFile,
29 |         FILE_MAP_ALL_ACCESS,
30 |         0, 0, size);
31 |
32 |     if (ptr == NULL) {
33 |         fprintf(stderr, "Error mapping shared memory (%lu)\n", GetLastError());
34 |         CloseHandle(hMapFile);
35 |         return NULL;

```

```

36     }
37
38     return ptr;
39 }
40
41 void* CpOpenSharedMemory(const char* name, size_t size) {
42     HANDLE hMapFile = OpenFileMapping(
43         FILE_MAP_ALL_ACCESS,
44         FALSE,
45         name);
46
47     if (hMapFile == NULL) {
48         fprintf(stderr, "Error opening shared memory (%lu)\n", GetLastError());
49         return NULL;
50     }
51
52     void* ptr = MapViewOfFile(
53         hMapFile,
54         FILE_MAP_ALL_ACCESS,
55         0, 0, size);
56
57     if (ptr == NULL) {
58         fprintf(stderr, "Error mapping shared memory (%lu)\n", GetLastError());
59         CloseHandle(hMapFile);
60         return NULL;
61     }
62
63     return ptr;
64 }
65
66 void CpCloseSharedMemory(const char* name, void* ptr, size_t size) {
67     if (ptr != NULL) {
68         UnmapViewOfFile(ptr);
69     }
70 }
71
72 int CpProcessCreate(process_t* proc, const char* path) {
73     if (!proc || !path) return -1;
74
75     SECURITY_ATTRIBUTES sa;
76     sa.nLength = sizeof(SECURITY_ATTRIBUTES);
77     sa.bInheritHandle = TRUE;
78     sa.lpSecurityDescriptor = NULL;
79
80     HANDLE childStdoutRead = NULL;
81     HANDLE childStdoutWrite = NULL;
82     HANDLE childStdinRead = NULL;
83     HANDLE childStdinWrite = NULL;
84
85     if (!CreatePipe(&childStdoutRead, &childStdoutWrite, &sa, 0)) return -1;
86     if (!CreatePipe(&childStdinRead, &childStdinWrite, &sa, 0)) {
87         CloseHandle(childStdoutRead);
88         CloseHandle(childStdoutWrite);
89         return -1;
90     }
91
92     SetHandleInformation(childStdoutRead, HANDLE_FLAG_INHERIT, 0);
93     SetHandleInformation(childStdinWrite, HANDLE_FLAG_INHERIT, 0);

```

```

94
95     STARTUPINFOA si;
96     PROCESS_INFORMATION pi;
97     ZeroMemory(&si, sizeof(si));
98     si.cb = sizeof(si);
99     si.hStdError = childStdoutWrite;
100    si.hStdOutput = childStdoutWrite;
101    si.hStdInput = childStdinRead;
102    si.dwFlags |= STARTF_USESTDHANDLES;
103
104    char cmdline[1024];
105    strncpy(cmdline, path, sizeof(cmdline)-1);
106    cmdline[sizeof(cmdline)-1] = '\0';
107
108    if (!CreateProcessA(NULL, cmdline, NULL, NULL, TRUE, 0, NULL, NULL, &si, &pi)) {
109        CloseHandle(childStdoutRead);
110        CloseHandle(childStdoutWrite);
111        CloseHandle(childStdinRead);
112        CloseHandle(childStdinWrite);
113        return -1;
114    }
115
116    CloseHandle(childStdoutWrite);
117    CloseHandle(childStdinRead);
118
119    proc->handle = pi.hProcess;
120    proc->stdinWrite = childStdinWrite;
121    proc->stdoutRead = childStdoutRead;
122
123    CloseHandle(pi.hThread);
124    return 0;
125 }
126
127 int CpProcessWrite(process_t* proc, const char* data, size_t size) {
128     if (!proc || !data) return -1;
129     DWORD written = 0;
130     if (!WriteFile(proc->stdinWrite, data, (DWORD)size, &written, NULL)) {
131         return -1;
132     }
133     return (int)written;
134 }
135
136 int CpProcessRead(process_t* proc, char* buffer, size_t size) {
137     if (!proc || !buffer || size == 0) return -1;
138     DWORD readBytes = 0;
139     if (!ReadFile(proc->stdoutRead, buffer, (DWORD)(size - 1), &readBytes, NULL)) {
140         return -1;
141     }
142     buffer[readBytes] = '\0';
143     return (int)readBytes;
144 }
145
146 int CpProcessClose(process_t* proc) {
147     if (!proc) return -1;
148     int exitCode = -1;
149     if (proc->stdinWrite) {
150         CloseHandle(proc->stdinWrite);
151         proc->stdinWrite = NULL;

```

```

152     }
153     if (proc->stdoutRead) {
154         CloseHandle(proc->stdoutRead);
155         proc->stdoutRead = NULL;
156     }
157     if (proc->handle) {
158         WaitForSingleObject(proc->handle, INFINITE);
159         DWORD code;
160         if (GetExitCodeProcess(proc->handle, &code)) {
161             exitCode = (int)code;
162         }
163         CloseHandle(proc->handle);
164         proc->handle = NULL;
165     }
166     return exitCode;
167 }
168
169 #else
170
171 #include <unistd.h>
172 #include <sys/wait.h>
173 #include <errno.h>
174
175 void* CpCreateSharedMemory(const char* name, size_t size) { return NULL; }
176 void* CpOpenSharedMemory(const char* name, size_t size) { return NULL; }
177 void CpCloseSharedMemory(const char* name, void* ptr, size_t size) {}
178
179 int CpProcessCreate(process_t* proc, const char* path) {
180     if (!proc || !path) return -1;
181     int inpipe[2];
182     int outpipe[2];
183
184     if (pipe(inpipe) == -1) return -1;
185     if (pipe(outpipe) == -1) {
186         close(inpipe[0]); close(inpipe[1]);
187         return -1;
188     }
189
190     pid_t pid = fork();
191     if (pid == -1) {
192         close(inpipe[0]); close(inpipe[1]);
193         close(outpipe[0]); close(outpipe[1]);
194         return -1;
195     }
196
197     if (pid == 0) {
198         dup2(inpipe[0], STDIN_FILENO);
199         dup2(outpipe[1], STDOUT_FILENO);
200
201         close(inpipe[0]); close(inpipe[1]);
202         close(outpipe[0]); close(outpipe[1]);
203
204         execl(path, path, (char*)NULL);
205         _exit(127);
206     } else {
207         close(inpipe[0]);
208         close(outpipe[1]);
209         proc->pid = pid;

```

```

210     proc->stdin_fd = inpipe[1];
211     proc->stdout_fd = outpipe[0];
212     return 0;
213 }
214 }
215
216 int CpProcessWrite(process_t* proc, const char* data, size_t size) {
217     if (!proc || !data) return -1;
218     ssize_t n = write(proc->stdin_fd, data, size);
219     if (n == -1) return -1;
220     return (int)n;
221 }
222
223 int CpProcessRead(process_t* proc, char* buffer, size_t size) {
224     if (!proc || !buffer || size == 0) return -1;
225     ssize_t n = read(proc->stdout_fd, buffer, (ssize_t)(size - 1));
226     if (n == -1) return -1;
227     if (n == 0) {
228         buffer[0] = '\0';
229         return 0;
230     }
231     buffer[n] = '\0';
232     return (int)n;
233 }
234
235 int CpProcessClose(process_t* proc) {
236     if (!proc) return -1;
237     int status = -1;
238     if (proc->stdin_fd != -1) {
239         close(proc->stdin_fd);
240         proc->stdin_fd = -1;
241     }
242     if (proc->stdout_fd != -1) {
243         close(proc->stdout_fd);
244         proc->stdout_fd = -1;
245     }
246     if (proc->pid > 0) {
247         waitpid(proc->pid, &status, 0);
248         if (WIFEXITED(status)) return WEXITSTATUS(status);
249     }
250     return -1;
251 }
252
253 #endif
254
255 size_t CpStringLength(const char* str) {
256     return str ? strlen(str) : 0;
257 }
258
259 int CpStringContains(const char* str, const char* substr) {
260     if (!str || !substr) return 0;
261     return strstr(str, substr) != NULL;
262 }

```

Листинг 3: *library.c*

```
1 | #include "library.h"
```

```

2  #include "shared_data.h"
3  #include <stdlib.h>
4  #include <stdio.h>
5  #include <string.h>
6  #include <ctype.h>
7
8  #define BUFFER_SIZE 1024
9
10 int main(void) {
11     process_t child;
12     char fileName[BUFFER_SIZE];
13     char inputBuffer[BUFFER_SIZE];
14
15     const char* shm_name = "Local\\DivisionLab3";
16     SharedData* shared_data = (SharedData*)CpCreateSharedMemory(shm_name, sizeof(
        SharedData));
17
18     if (shared_data == NULL) {
19         fprintf(stderr, "Error: Failed to create shared memory\n");
20         return EXIT_FAILURE;
21     }
22
23     memset(shared_data, 0, sizeof(SharedData));
24     shared_data->data_ready = 0;
25     shared_data->processing_complete = 1;
26     shared_data->division_by_zero = 0;
27     shared_data->terminate_process = 0;
28     shared_data->numbers_count = 0;
29
30     printf("Enter file name: ");
31     if (!fgets(fileName, sizeof(fileName), stdin)) {
32         fprintf(stderr, "Error: failed to read file name\n");
33         CpCloseSharedMemory(shm_name, shared_data, sizeof(SharedData));
34         return EXIT_FAILURE;
35     }
36     TrimNewline(fileName);
37     if (strlen(fileName) == 0) {
38         fprintf(stderr, "Error: empty file name\n");
39         CpCloseSharedMemory(shm_name, shared_data, sizeof(SharedData));
40         return EXIT_FAILURE;
41     }
42
43     strncpy(shared_data->filename, fileName, FILENAME_LENGTH - 1);
44     shared_data->filename[FILENAME_LENGTH - 1] = '\0';
45
46     char cmdLine[BUFFER_SIZE * 2];
47     snprintf(cmdLine, sizeof(cmdLine), "child.exe \"%s\"", fileName);
48
49     if (CpProcessCreate(&child, cmdLine) != 0) {
50         fprintf(stderr, "Error: failed to create child process\n");
51         CpCloseSharedMemory(shm_name, shared_data, sizeof(SharedData));
52         return EXIT_FAILURE;
53     }
54
55     printf("Child process created. Using FILE MAPPING for IPC.\n");
56     printf("Enter numbers separated by spaces (cumulative division):\n");
57     printf("For exit write 'exit' or empty line\n");
58

```

```

59 while (1) {
60     printf("> ");
61     fflush(stdout);
62
63     if (!fgets(inputBuffer, sizeof(inputBuffer), stdin)) {
64         break;
65     }
66     TrimNewline(inputBuffer);
67
68     if (strcmp(inputBuffer, "exit") == 0 || strlen(inputBuffer) == 0) {
69         shared_data->terminate_process = 1;
70         shared_data->data_ready = 1;
71         break;
72     }
73
74     int numbers[MAX_NUMBERS];
75     int count = 0;
76     char* token = strtok(inputBuffer, " \t");
77
78     while (token != NULL && count < MAX_NUMBERS) {
79         int valid = 1;
80         for (int i = 0; token[i] != '\0'; i++) {
81             if (!isdigit(token[i]) && !(i == 0 && token[i] == '-')) {
82                 valid = 0;
83                 break;
84             }
85         }
86
87         if (valid) {
88             numbers[count++] = atoi(token);
89         }
90         token = strtok(NULL, " \t");
91     }
92
93     if (count < 2) {
94         printf("Error: need at least 2 numbers\n");
95         continue;
96     }
97
98     shared_data->numbers_count = count;
99     for (int i = 0; i < count; i++) {
100         shared_data->numbers[i] = numbers[i];
101     }
102
103     shared_data->data_ready = 1;
104     shared_data->processing_complete = 0;
105
106     printf("Waiting for child to process...\n");
107     while (!shared_data->processing_complete && !shared_data->division_by_zero) {
108         Sleep(10);
109     }
110
111     if (shared_data->division_by_zero) {
112         printf("EMERGENCY: Division by zero detected! Shutting down...\n");
113         break;
114     }
115
116     printf("Processing complete. Results written to file.\n");

```

```

117     }
118
119     // Cleanup
120     printf("Shutting down...\n");
121     int exitCode = CpProcessClose(&child);
122     CpCloseSharedMemory(shm_name, shared_data, sizeof(SharedData));
123
124     printf("Parent process finished. Child exit code: %d\n", exitCode);
125     return EXIT_SUCCESS;
126 }

```

Листинг 4: *parent.c*

```

1  #include "library.h"
2  #include "shared_data.h"
3  #include <stdio.h>
4  #include <stdlib.h>
5  #include <string.h>
6  #include <ctype.h>
7
8  #ifdef _WIN32
9  #include <windows.h>
10 #endif
11
12 #define BUFFER_SIZE 1024
13
14 int main(int argc, char* argv[]) {
15     FILE* output_file = NULL;
16
17     const char* shm_name = "Local\\DivisionLab3";
18     SharedData* shared_data = (SharedData*)CpOpenSharedMemory(shm_name, sizeof(
19         SharedData));
20
21     if (shared_data == NULL) {
22         fprintf(stderr, "Error: Failed to open shared memory\n");
23         return EXIT_FAILURE;
24     }
25
26     if (argc != 2) {
27         fprintf(stderr, "Usage: %s <filename>\n", argv[0]);
28         CpCloseSharedMemory(shm_name, shared_data, sizeof(SharedData));
29         return EXIT_FAILURE;
30     }
31
32     const char* filename = argv[1];
33
34     output_file = fopen(filename, "w");
35     if (output_file == NULL) {
36         fprintf(stderr, "Error: Cannot open file '%s'\n", filename);
37         CpCloseSharedMemory(shm_name, shared_data, sizeof(SharedData));
38         return EXIT_FAILURE;
39     }
40
41     fprintf(output_file, "Child process started. Output file: %s\n", filename);
42     fprintf(output_file, "Using FILE MAPPING for IPC with parent\n");
43     fflush(output_file);

```



```

44     printf("Child process started. Monitoring shared memory...\n");
45
46     while (!shared_data->terminate_process) {
47         if (shared_data->data_ready) {
48             if (shared_data->numbers_count < 2) {
49                 fprintf(output_file, "Error: need at least 2 numbers, got %d\n",
50                     shared_data->numbers_count);
51             } else {
52                 fprintf(output_file, "Input: ");
53                 for (int i = 0; i < shared_data->numbers_count; i++) {
54                     fprintf(output_file, "%d ", shared_data->numbers[i]);
55                 }
56                 fprintf(output_file, "\n");
57
58                 int division_by_zero = 0;
59                 for (int i = 1; i < shared_data->numbers_count; i++) {
60                     if (shared_data->numbers[i] == 0) {
61                         fprintf(output_file, "ERROR: Division by zero! %d / %d\n",
62                             shared_data->numbers[0], shared_data->numbers[i]);
63                         division_by_zero = 1;
64                         shared_data->division_by_zero = 1;
65                         break;
66                     }
67                 }
68
69                 if (!division_by_zero) {
70                     float result = (float)shared_data->numbers[0];
71                     fprintf(output_file, "Division results:\n");
72                     for (int i = 1; i < shared_data->numbers_count; i++) {
73                         float temp = result / (float)shared_data->numbers[i];
74                         fprintf(output_file, " %.2f / %.2f = %.2f\n",
75                             result, (float)shared_data->numbers[i], temp);
76                         result = temp;
77                     }
78                     fprintf(output_file, "Final result: %.2f\n", result);
79                     fprintf(output_file, "---\n");
80                 }
81             }
82
83             fflush(output_file);
84             shared_data->processing_complete = 1;
85             shared_data->data_ready = 0;
86
87             if (shared_data->division_by_zero) {
88                 fprintf(output_file, "EMERGENCY SHUTDOWN: Division by zero detected\n");
89                 ;
90                 fflush(output_file);
91                 fclose(output_file);
92                 CpCloseSharedMemory(shm_name, shared_data, sizeof(SharedData));
93
94                 #ifdef _WIN32
95                     ExitProcess(2);
96                 #else
97                     exit(2);
98                 #endif
99             }
100         }
101     }

```

```

100     #ifdef _WIN32
101         Sleep(10);
102     #else
103         usleep(10000);
104     #endif
105 }
106
107 fprintf(output_file, "Child process finished normally\n");
108 fclose(output_file);
109 CpCloseSharedMemory(shm_name, shared_data, sizeof(SharedData));
110
111 printf("Child process finished\n");
112 return EXIT_SUCCESS;
113 }

```

Листинг 5: *child.c*

Системные вызовы

```

execve("./parent.exe", ["/parent.exe"], 0x7ffee5114460 /* 26 vars */) = 0
mmap(NULL, 584, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x783f73044000
arch_prctl(ARCH_SET_FS, 0x783f73044180) = 0
set_tid_address(0x4a24f8) = 437
gettid() = 437
gettid() = 437
gettid() = 437
gettid() = 437
brk(NULL) = 0x189fc000
brk(0x189fe000) = 0x189fe000
mmap(0x189fc000, 4096, PROT_NONE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x189fc000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x783f73043000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x783f73042000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x783f73041000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x783f73040000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x783f7303f000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x783f7303e000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x783f7303d000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x783f7303c000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x783f7303a000
sched_getaffinity(0, 128, [0 1 2 3 4 5 6 7]) = 8
getpid() = 437
getpid() = 437
uname({sysname="Linux", nodename="DESKTOP-PDVOEPV", ...}) = 0
getcwd("/mnt/d/MAI-year2/os/os_lab3/build", 4096) = 34
readlink("parent.exe", 0x7ffddcec7610, 4096) = -1 EINVAL (Invalid argument)
getcwd("/mnt/d/MAI-year2/os/os_lab3/build", 4097) = 34
open("/proc/self/mountinfo", O_RDONLY|O_LARGEFILE) = 3
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x783f73039000
read(3, "165 170 0:29 / /usr/lib/modules/"..., 1024) = 1024
read(3, "suid,relatime shared:38 -devtmp"..., 1024) = 1024

```

```

read(3, "lowerdir=/systemvhd,upperdir=/s"... ,1024) = 1024
read(3, "nosuid,nodev,noexec,relatime sh"... ,1024) = 545
read(3, "",1024) = 0
close(3) = 0
munmap(0x783f73039000,4096) = 0
mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x783f73038000
mmap(NULL,16384,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x783f73034000
getcwd("/mnt/d/MAI-year2/os/os_lab3/build",4096) = 34
munmap(0x783f73034000,16384) = 0
getcwd("/mnt/d/MAI-year2/os/os_lab3/build",4096) = 34
open("/proc/self/mountinfo",O_RDONLY|O_LARGEFILE) = 3
mmap(NULL,4096,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x783f73037000
read(3, "165 170 0:29 / /usr/lib/modules/"... ,1024) = 1024
read(3, "suid,relatime shared:38 -devtmp"... ,1024) = 1024
read(3, "lowerdir=/systemvhd,upperdir=/s"... ,1024) = 1024
read(3, "nosuid,nodev,noexec,relatime sh"... ,1024) = 545
read(3, "",1024) = 0
close(3) = 0
munmap(0x783f73037000,4096) = 0
ioctl(0,TCGETS,{c_iflag=ICRNL|IXON,c_oflag=NLO|CRO|TABO|BSO|VTO|FFO|OPOST|ONLCR,c_cfl
= 0
ioctl(1,TCGETS,{c_iflag=ICRNL|IXON,c_oflag=NLO|CRO|TABO|BSO|VTO|FFO|OPOST|ONLCR,c_cfl
= 0
ioctl(2,TCGETS,{c_iflag=ICRNL|IXON,c_oflag=NLO|CRO|TABO|BSO|VTO|FFO|OPOST|ONLCR,c_cfl
= 0
ioctl(0,TIOCGPGRP,[434]) = 0
getpgid(0) = 434
fstat(0,{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0),...}) = 0
fstat(1,{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0),...}) = 0
fstat(2,{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0),...}) = 0
ioctl(0,TIOCGWINSZ,{ws_row=30,ws_col=120,ws_xpixel=0,ws_ypixel=0}) = 0
ioctl(0,TCSETS,{c_iflag=,c_oflag=NLO|CRO|TABO|BSO|VTO|FFO|ONLCR,c_cflag=B38400|CS8|CR
= 0
dup(0) = 3
munmap(0x783f73038000,8192) = 0
socket(AF_VSOCK,SOCK_STREAM|SOCK_CLOEXEC,0) = 4
bind(4,{sa_family=AF_VSOCK,svm_cid=VMADDR_CID_ANY,svm_port=VMADDR_PORT_ANY,svm_flags=
= 0
getsockname(4,{sa_family=AF_VSOCK,svm_cid=VMADDR_CID_ANY,svm_port=0xffbaac7,svm_flags
= 0
listen(4,4) = 0
access("/run/WSL/290_interop",F_OK) = 0
socket(AF_UNIX,SOCK_STREAM|SOCK_CLOEXEC,0) = 5
connect(5,{sa_family=AF_UNIX,sun_path="/run/WSL/290_interop"},110) = 0
mmap(NULL,16384,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x783f73036000
open("/proc/cmdline",O_RDONLY|O_LARGEFILE) = 6
mmap(NULL,4096,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x783f73035000
readv(6, [{iov_base="initrd=\\initrd.img WSL_ROOT_INIT"... ,iov_len=4095},{iov_base="",

```

```
= 145
readv(6,[{iov_base="",iov_len=3950},{iov_base="",iov_len=1024}],2) = 0
lseek(6,0,SEEK_CUR) = 145
close(6) = 0
munmap(0x783f73035000,4096) = 0
munmap(0x783f73036000,16384) = 0
write(5,"\10\0\0\0\235\0\0\0\1\0\0\0\307\252\373\17\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0".
= 157
poll([fd=4,events=POLLIN],1,10000) = 1 ([fd=4,revents=POLLIN])
accept4(4,{sa_family=AF_VSOCK,svm_cid=VMADDR_CID_HOST,svm_port=0xc373d0cf,svm_flags=0
= 6
poll([fd=4,events=POLLIN],1,10000) = 1 ([fd=4,revents=POLLIN])
accept4(4,{sa_family=AF_VSOCK,svm_cid=VMADDR_CID_HOST,svm_port=0xc373d0d0,svm_flags=0
= 7
poll([fd=4,events=POLLIN],1,10000) = 1 ([fd=4,revents=POLLIN])
accept4(4,{sa_family=AF_VSOCK,svm_cid=VMADDR_CID_HOST,svm_port=0xc373d0d1,svm_flags=0
= 8
poll([fd=4,events=POLLIN],1,10000) = 1 ([fd=4,revents=POLLIN])
accept4(4,{sa_family=AF_VSOCK,svm_cid=VMADDR_CID_HOST,svm_port=0xc373d0d2,svm_flags=0
= 9
close(4) = 0
rt_sigprocmask(SIG_BLOCK,[INT WINCH],NULL,8) = 0
signalfd4(-1,[INT WINCH],8,0) = 4
poll([fd=0,events=POLLIN],[fd=7,events=POLLIN],[fd=8,events=POLLIN],[fd=9,events=POLL
= 1 ([fd=9,revents=POLLIN])
recvfrom(9,"\v\0\0\0 \0\0\0\0\0\0\0",12,MSG_WAITALL,NULL,NULL) = 12
recvfrom(9,"\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0",20,0,NULL,NULL) = 20
poll([fd=0,events=POLLIN],[fd=7,events=POLLIN],[fd=8,events=POLLIN],[fd=9,events=POLL
= 1 ([fd=7,revents=POLLIN])
mmap(NULL,16384,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x783f73036000
read(7,"\33[6n",4096) = 4
write(1,"\33[6n",4) = 4
poll([fd=0,events=POLLIN],[fd=7,events=POLLIN],[fd=8,events=POLLIN],[fd=9,events=POLL
= 1 ([fd=0,revents=POLLIN])
read(0,"\33[30;1R",4096) = 7
write(6,"\33[30;1R",7) = 7
poll([fd=0,events=POLLIN],[fd=7,events=POLLIN],[fd=8,events=POLLIN],[fd=9,events=POLL
= 1 ([fd=7,revents=POLLIN])
read(7,"\33[?9001h\33[?1004h",4096) = 16
write(1,"\33[?9001h\33[?1004h",16) = 16
poll([fd=0,events=POLLIN],[fd=7,events=POLLIN],[fd=8,events=POLLIN],[fd=9,events=POLL
= 1 ([fd=7,revents=POLLIN])
read(7,"\33[m\33]0;D:\\MAI-year2\\os\\os_lab3\\b"... ,4096) = 54
write(1,"\33[m\33]0;D:\\MAI-year2\\os\\os_lab3\\b"... ,54) = 54
poll([fd=0,events=POLLIN],[fd=7,events=POLLIN],[fd=8,events=POLLIN],[fd=9,events=POLL
= 1 ([fd=7,revents=POLLIN])
read(7,"Enter file name: ",4096) = 17
write(1,"Enter file name: ",17) = 17
```

```

poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[83;31;115;1;32;1_",4096) = 19
write(6,"\33[83;31;115;1;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=7,revents=POLLIN}])
read(7,"s",4096) = 1
write(1,"s",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[83;31;115;0;32;1_",4096) = 19
write(6,"\33[83;31;115;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[84;20;116;1;32;1_",4096) = 19
write(6,"\33[84;20;116;1;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=7,revents=POLLIN}])
read(7,"t",4096) = 1
write(1,"t",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[84;20;116;0;32;1_",4096) = 19
write(6,"\33[84;20;116;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[8;14;8;1;32;1_",4096) = 16
write(6,"\33[8;14;8;1;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=7,revents=POLLIN}])
read(7,"\10 \10",4096) = 3
write(1,"\10 \10",3) = 3
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[8;14;8;0;32;1_",4096) = 16
write(6,"\33[8;14;8;0;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[8;14;8;1;32;1_",4096) = 16
write(6,"\33[8;14;8;1;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=7,revents=POLLIN}])
read(7,"\10 \10",4096) = 3
write(1,"\10 \10",3) = 3
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[8;14;8;0;32;1_",4096) = 16
write(6,"\33[8;14;8;0;32;1_",16) = 16

```

```

poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[84;20;116;1;32;1_",4096) = 19
write(6,"\33[84;20;116;1;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=7,revents=POLLIN}])
read(7,"t",4096) = 1
write(1,"t",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[84;20;116;0;32;1_",4096) = 19
write(6,"\33[84;20;116;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[82;19;114;1;32;1_",4096) = 19
write(6,"\33[82;19;114;1;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=7,revents=POLLIN}])
read(7,"r",4096) = 1
write(1,"r",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[82;19;114;0;32;1_",4096) = 19
write(6,"\33[82;19;114;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[65;16;97;1;32;1_",4096) = 18
write(6,"\33[65;16;97;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=7,revents=POLLIN}])
read(7,"a",4096) = 1
write(1,"a",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[67;46;99;1;32;1_",4096) = 18
write(6,"\33[67;46;99;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 2 ([{fd=0,revents=POLLIN},{fd=7,revents=POLLIN}])
read(0,"\33[65;16;97;0;32;1_",4096) = 18
write(6,"\33[65;16;97;0;32;1_",18) = 18
read(7,"c",4096) = 1
write(1,"c",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[67;46;99;0;32;1_",4096) = 18
write(6,"\33[67;46;99;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])

```

```

read(0, "\33[69;18;101;1;32;1_", 4096) = 19
write(6, "\33[69;18;101;1;32;1_", 19) = 19
poll([{fd=0, events=POLLIN}, {fd=7, events=POLLIN}, {fd=8, events=POLLIN}, {fd=9, events=POLLIN}], 4, &revents) = 1 ([{fd=7, revents=POLLIN}])
read(7, "e", 4096) = 1
write(1, "e", 1) = 1
poll([{fd=0, events=POLLIN}, {fd=7, events=POLLIN}, {fd=8, events=POLLIN}, {fd=9, events=POLLIN}], 4, &revents) = 1 ([{fd=0, revents=POLLIN}])
read(0, "\33[69;18;101;0;32;1_", 4096) = 19
write(6, "\33[69;18;101;0;32;1_", 19) = 19
poll([{fd=0, events=POLLIN}, {fd=7, events=POLLIN}, {fd=8, events=POLLIN}, {fd=9, events=POLLIN}], 4, &revents) = 1 ([{fd=0, revents=POLLIN}])
read(0, "\33[16;54;0;1;48;1_", 4096) = 17
write(6, "\33[16;54;0;1;48;1_", 17) = 17
poll([{fd=0, events=POLLIN}, {fd=7, events=POLLIN}, {fd=8, events=POLLIN}, {fd=9, events=POLLIN}], 4, &revents) = 1 ([{fd=0, revents=POLLIN}])
read(0, "\33[190;51;46;1;48;1_", 4096) = 19
write(6, "\33[190;51;46;1;48;1_", 19) = 19
poll([{fd=0, events=POLLIN}, {fd=7, events=POLLIN}, {fd=8, events=POLLIN}, {fd=9, events=POLLIN}], 4, &revents) = 1 ([{fd=7, revents=POLLIN}])
read(7, ".", 4096) = 1
write(1, ".", 1) = 1
poll([{fd=0, events=POLLIN}, {fd=7, events=POLLIN}, {fd=8, events=POLLIN}, {fd=9, events=POLLIN}], 4, &revents) = 1 ([{fd=0, revents=POLLIN}])
read(0, "\33[16;54;0;0;32;1_", 4096) = 17
write(6, "\33[16;54;0;0;32;1_", 17) = 17
poll([{fd=0, events=POLLIN}, {fd=7, events=POLLIN}, {fd=8, events=POLLIN}, {fd=9, events=POLLIN}], 4, &revents) = 1 ([{fd=0, revents=POLLIN}])
read(0, "\33[190;51;59;0;32;1_", 4096) = 19
write(6, "\33[190;51;59;0;32;1_", 19) = 19
poll([{fd=0, events=POLLIN}, {fd=7, events=POLLIN}, {fd=8, events=POLLIN}, {fd=9, events=POLLIN}], 4, &revents) = 1 ([{fd=0, revents=POLLIN}])
read(0, "\33[84;20;116;1;32;1_", 4096) = 19
write(6, "\33[84;20;116;1;32;1_", 19) = 19
poll([{fd=0, events=POLLIN}, {fd=7, events=POLLIN}, {fd=8, events=POLLIN}, {fd=9, events=POLLIN}], 4, &revents) = 1 ([{fd=7, revents=POLLIN}])
read(7, "t", 4096) = 1
write(1, "t", 1) = 1
poll([{fd=0, events=POLLIN}, {fd=7, events=POLLIN}, {fd=8, events=POLLIN}, {fd=9, events=POLLIN}], 4, &revents) = 1 ([{fd=0, revents=POLLIN}])
read(0, "\33[84;20;116;0;32;1_", 4096) = 19
write(6, "\33[84;20;116;0;32;1_", 19) = 19
poll([{fd=0, events=POLLIN}, {fd=7, events=POLLIN}, {fd=8, events=POLLIN}, {fd=9, events=POLLIN}], 4, &revents) = 1 ([{fd=0, revents=POLLIN}])
read(0, "\33[88;45;120;1;32;1_", 4096) = 19
write(6, "\33[88;45;120;1;32;1_", 19) = 19
poll([{fd=0, events=POLLIN}, {fd=7, events=POLLIN}, {fd=8, events=POLLIN}, {fd=9, events=POLLIN}], 4, &revents) = 1 ([{fd=7, revents=POLLIN}])

```

```

read(7,"x",4096)                = 1
write(1,"x",1)                   = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=0,events=POLLIN}])
read(0,"\33[84;20;116;1;32;1_",4096) = 19
write(6,"\33[84;20;116;1;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=7,events=POLLIN}])
read(7,"t",4096)                = 1
write(1,"t",1)                   = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=0,events=POLLIN}])
read(0,"\33[88;45;120;0;32;1_",4096) = 19
write(6,"\33[88;45;120;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=0,events=POLLIN}])
read(0,"\33[84;20;116;0;32;1_",4096) = 19
write(6,"\33[84;20;116;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=0,events=POLLIN}])
read(0,"\33[8;14;8;1;32;1_",4096) = 16
write(6,"\33[8;14;8;1;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=7,events=POLLIN}])
read(7,"\10 \10",4096)          = 3
write(1,"\10 \10",3)            = 3
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=0,events=POLLIN}])
read(0,"\33[8;14;8;1;32;1_",4096) = 16
write(6,"\33[8;14;8;1;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=7,events=POLLIN}])
read(7,"\10 \10",4096)          = 3
write(1,"\10 \10",3)            = 3
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=0,events=POLLIN}])
read(0,"\33[8;14;8;1;32;1_",4096) = 16
write(6,"\33[8;14;8;1;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=7,events=POLLIN}])
read(7,"\10 \10",4096)          = 3
write(1,"\10 \10",3)            = 3
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=0,events=POLLIN}])
read(0,"\33[8;14;8;1;32;1_",4096) = 16
write(6,"\33[8;14;8;1;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=7,events=POLLIN}])

```


[illegible]

[illegible]

[illegible]

```

read(0, "\33[84;20;116;1;32;1_", 4096) = 19
write(6, "\33[84;20;116;1;32;1_", 19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}], 1000, 0) = 1 ([{fd=7,events=POLLIN}])
read(7, "t", 4096) = 1
write(1, "t", 1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}], 1000, 0) = 1 ([{fd=0,events=POLLIN}])
read(0, "\33[84;20;116;0;32;1_", 4096) = 19
write(6, "\33[84;20;116;0;32;1_", 19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}], 1000, 0) = 1 ([{fd=0,events=POLLIN}])
read(0, "\33[69;18;101;1;32;1_", 4096) = 19
write(6, "\33[69;18;101;1;32;1_", 19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}], 1000, 0) = 1 ([{fd=7,events=POLLIN}])
read(7, "e", 4096) = 1
write(1, "e", 1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}], 1000, 0) = 1 ([{fd=0,events=POLLIN}])
read(0, "\33[69;18;101;0;32;1_", 4096) = 19
write(6, "\33[69;18;101;0;32;1_", 19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}], 1000, 0) = 1 ([{fd=0,events=POLLIN}])
read(0, "\33[83;31;115;1;32;1_", 4096) = 19
write(6, "\33[83;31;115;1;32;1_", 19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}], 1000, 0) = 1 ([{fd=7,events=POLLIN}])
read(7, "s", 4096) = 1
write(1, "s", 1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}], 1000, 0) = 1 ([{fd=0,events=POLLIN}])
read(0, "\33[84;20;116;1;32;1_", 4096) = 19
write(6, "\33[84;20;116;1;32;1_", 19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}], 1000, 0) = 2 ([{fd=0,events=POLLIN},{fd=7,events=POLLIN}])
read(0, "\33[83;31;115;0;32;1_", 4096) = 19
write(6, "\33[83;31;115;0;32;1_", 19) = 19
read(7, "t", 4096) = 1
write(1, "t", 1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}], 1000, 0) = 1 ([{fd=0,events=POLLIN}])
read(0, "\33[84;20;116;0;32;1_", 4096) = 19
write(6, "\33[84;20;116;0;32;1_", 19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}], 1000, 0) = 1 ([{fd=0,events=POLLIN}])
read(0, "\33[97;79;49;1;32;1_", 4096) = 18
write(6, "\33[97;79;49;1;32;1_", 18) = 18

```

```

poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096)
= 1 ([{fd=7,events=POLLIN}])
read(7,"1",4096) = 1
write(1,"1",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[97;79;49;0;32;1_",4096) = 18
write(6,"\33[97;79;49;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[13;28;13;1;32;1_",4096) = 18
write(6,"\33[13;28;13;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096)
= 1 ([{fd=7,events=POLLIN}])
read(7,"\r",4096) = 1
write(1,"\r",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096)
= 1 ([{fd=7,events=POLLIN}])
read(7,"\n",4096) = 1
write(1,"\n",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096)
= 1 ([{fd=7,events=POLLIN}])
read(7,"Child process created. Using FILE"...,4096) = 51
write(1,"Child process created. Using FILE"...,51) = 51
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096)
= 1 ([{fd=7,events=POLLIN}])
read(7,"\n",4096) = 1
write(1,"\n",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096)
= 1 ([{fd=7,events=POLLIN}])
read(7,"Enter numbers separated by space"...,4096) = 57
write(1,"Enter numbers separated by space"...,57) = 57
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096)
= 1 ([{fd=7,events=POLLIN}])
read(7,"\nFor exit write 'exit'or empty "...,4096) = 37
write(1,"\nFor exit write 'exit'or empty "...,37) = 37
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096)
= 1 ([{fd=7,events=POLLIN}])
read(7,"\n>\33[1C",4096) = 6
write(1,"\n>\33[1C",6) = 6
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[13;28;13;0;32;1_",4096) = 18
write(6,"\33[13;28;13;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[100;75;52;1;32;1_",4096) = 19
write(6,"\33[100;75;52;1;32;1_",19) = 19

```

```

poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,events=POLLIN}])
read(7,"4",4096) = 1
write(1,"4",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[100;75;52;0;32;1_",4096) = 19
write(6,"\33[100;75;52;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[96;82;48;1;32;1_",4096) = 18
write(6,"\33[96;82;48;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,events=POLLIN}])
read(7,"0",4096) = 1
write(1,"0",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[96;82;48;0;32;1_",4096) = 18
write(6,"\33[96;82;48;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[96;82;48;1;32;1_",4096) = 18
write(6,"\33[96;82;48;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,events=POLLIN}])
read(7,"0",4096) = 1
write(1,"0",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[96;82;48;0;32;1_",4096) = 18
write(6,"\33[96;82;48;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[32;57;32;1;32;1_",4096) = 18
write(6,"\33[32;57;32;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,events=POLLIN}])
read(7," ",4096) = 1
write(1," ",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[101;76;53;1;32;1_",4096) = 19
write(6,"\33[101;76;53;1;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,events=POLLIN}])
read(7,"5",4096) = 1
write(1,"5",1) = 1

```

```

poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[32;57;32;0;32;1_",4096) = 18
write(6,"\33[32;57;32;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[101;76;53;0;32;1_",4096) = 19
write(6,"\33[101;76;53;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[96;82;48;1;32;1_",4096) = 18
write(6,"\33[96;82;48;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,revents=POLLIN}])
read(7,"0",4096) = 1
write(1,"0",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[96;82;48;0;32;1_",4096) = 18
write(6,"\33[96;82;48;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[8;14;8;1;32;1_",4096) = 16
write(6,"\33[8;14;8;1;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,revents=POLLIN}])
read(7,"\10 \10",4096) = 3
write(1,"\10 \10",3) = 3
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[8;14;8;0;32;1_",4096) = 16
write(6,"\33[8;14;8;0;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[8;14;8;1;32;1_",4096) = 16
write(6,"\33[8;14;8;1;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,revents=POLLIN}])
read(7,"\10 \10",4096) = 3
write(1,"\10 \10",3) = 3
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[8;14;8;0;32;1_",4096) = 16
write(6,"\33[8;14;8;0;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[105;73;57;1;32;1_",4096) = 19
write(6,"\33[105;73;57;1;32;1_",19) = 19

```

```

poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,revents=POLLIN}])
read(7,"9",4096) = 1
write(1,"9",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[105;73;57;0;32;1_",4096) = 19
write(6,"\33[105;73;57;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[8;14;8;1;32;1_",4096) = 16
write(6,"\33[8;14;8;1;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,revents=POLLIN}])
read(7,"\10 \10",4096) = 3
write(1,"\10 \10",3) = 3
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[8;14;8;0;32;1_",4096) = 16
write(6,"\33[8;14;8;0;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[102;77;54;1;32;1_",4096) = 19
write(6,"\33[102;77;54;1;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,revents=POLLIN}])
read(7,"6",4096) = 1
write(1,"6",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[102;77;54;0;32;1_",4096) = 19
write(6,"\33[102;77;54;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[96;82;48;1;32;1_",4096) = 18
write(6,"\33[96;82;48;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,revents=POLLIN}])
read(7,"0",4096) = 1
write(1,"0",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[96;82;48;0;32;1_",4096) = 18
write(6,"\33[96;82;48;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[32;57;32;1;32;1_",4096) = 18
write(6,"\33[32;57;32;1;32;1_",18) = 18

```



```

poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=7,revents=POLLIN}])
read(7," ",4096) = 1
write(1," ",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[32;57;32;0;32;1_",4096) = 18
write(6,"\33[32;57;32;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[105;73;57;1;32;1_",4096) = 19
write(6,"\33[105;73;57;1;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=7,revents=POLLIN}])
read(7,"9",4096) = 1
write(1,"9",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[105;73;57;0;32;1_",4096) = 19
write(6,"\33[105;73;57;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[13;28;13;1;32;1_",4096) = 18
write(6,"\33[13;28;13;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=7,revents=POLLIN}])
read(7,"\r",4096) = 1
write(1,"\r",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=7,revents=POLLIN}])
read(7,"\n",4096) = 1
write(1,"\n",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=7,revents=POLLIN}])
read(7,"Waiting for child to process...\r",4096) = 32
write(1,"Waiting for child to process...\r",32) = 32
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=7,revents=POLLIN}])
read(7,"\n",4096) = 1
write(1,"\n",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=7,revents=POLLIN}])
read(7,"Processing complete. Results written...",4096) = 46
write(1,"Processing complete. Results written...",46) = 46
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4096) = 1 ([{fd=7,revents=POLLIN}])
read(7,"\n",4096) = 1
write(1,"\n",1) = 1

```

```

poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=7,revents=POLLIN}])
read(7,">",4096) = 2
write(1,">",2) = 2
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[13;28;13;0;32;1_",4096) = 18
write(6,"\33[13;28;13;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[101;76;53;1;32;1_",4096) = 19
write(6,"\33[101;76;53;1;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=7,revents=POLLIN}])
read(7,"5",4096) = 1
write(1,"5",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[101;76;53;0;32;1_",4096) = 19
write(6,"\33[101;76;53;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[96;82;48;1;32;1_",4096) = 18
write(6,"\33[96;82;48;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=7,revents=POLLIN}])
read(7,"0",4096) = 1
write(1,"0",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[96;82;48;0;32;1_",4096) = 18
write(6,"\33[96;82;48;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[96;82;48;1;32;1_",4096) = 18
write(6,"\33[96;82;48;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=7,revents=POLLIN}])
read(7,"0",4096) = 1
write(1,"0",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[96;82;48;0;32;1_",4096) = 18
write(6,"\33[96;82;48;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}])
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[32;57;32;1;32;1_",4096) = 18
write(6,"\33[32;57;32;1;32;1_",18) = 18

```

```

poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,events=POLLIN}])
read(7," ",4096) = 1
write(1," ",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[32;57;32;0;32;1_",4096) = 18
write(6,"\33[32;57;32;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[0",4096) = 3
write(6,"\33[0",3) = 3
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[I",4096) = 3
write(6,"\33[I",3) = 3
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[103;71;55;1;32;1_",4096) = 19
write(6,"\33[103;71;55;1;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,events=POLLIN}])
read(7,"7",4096) = 1
write(1,"7",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[103;71;55;0;32;1_",4096) = 19
write(6,"\33[103;71;55;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[32;57;32;1;32;1_",4096) = 18
write(6,"\33[32;57;32;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,events=POLLIN}])
read(7," ",4096) = 1
write(1," ",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[32;57;32;0;32;1_",4096) = 18
write(6,"\33[32;57;32;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,events=POLLIN}])
read(0,"\33[96;82;48;1;32;1_",4096) = 18
write(6,"\33[96;82;48;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,events=POLLIN}])
read(7,"0",4096) = 1
write(1,"0",1) = 1

```

```

poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[96;82;48;0;32;1_",4096) = 18
write(6,"\33[96;82;48;0;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[220;43;42;1;32;1_",4096) = 19
write(6,"\33[220;43;42;1;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,revents=POLLIN}])
read(7,"*",4096) = 1
write(1,"*",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[220;43;42;0;32;1_",4096) = 19
write(6,"\33[220;43;42;0;32;1_",19) = 19
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[8;14;8;1;32;1_",4096) = 16
write(6,"\33[8;14;8;1;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,revents=POLLIN}])
read(7,"\10 \10",4096) = 3
write(1,"\10 \10",3) = 3
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[8;14;8;0;32;1_",4096) = 16
write(6,"\33[8;14;8;0;32;1_",16) = 16
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=0,revents=POLLIN}])
read(0,"\33[13;28;13;1;32;1_",4096) = 18
write(6,"\33[13;28;13;1;32;1_",18) = 18
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,revents=POLLIN}])
read(7,"\r",4096) = 1
write(1,"\r",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,revents=POLLIN}])
read(7,"\n",4096) = 1
write(1,"\n",1) = 1
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,revents=POLLIN}])
read(7,"Waiting for child to process...\r",4096) = 32
write(1,"Waiting for child to process...\r",32) = 32
poll([{fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}],4,1)
= 1 ([{fd=7,revents=POLLIN}])
read(7,"\n",4096) = 1
write(1,"\n",1) = 1

```

```

poll([fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}]
= 1 ([fd=7,revents=POLLIN]))
read(7,"EMERGENCY: Division by zero detected... ",4096) = 55
write(1,"EMERGENCY: Division by zero detected... ",55) = 55
poll([fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}]
= 1 ([fd=7,revents=POLLIN]))
read(7,"\n",4096) = 1
write(1,"\n",1) = 1
poll([fd=0,events=POLLIN},{fd=7,events=POLLIN},{fd=8,events=POLLIN},{fd=9,events=POLLIN}]
= 3 ([fd=7,revents=POLLIN},{fd=8,revents=POLLIN},{fd=9,revents=POLLIN}])
read(7,"Shutting down...\r",4096) = 17
write(1,"Shutting down...\r",17) = 17
read(8,"",4096) = 0
recvfrom(9,"\t\0\0\0\20\0\0\0\260\366\237\252",12,MSG_WAITALL,NULL,NULL) =
12
recvfrom(9,"\0\0\0\0",4,0,NULL,NULL) = 4
shutdown(6,SHUT_WR) = 0
poll([fd=-1},{fd=7,events=POLLIN},{fd=-1},{fd=-1},{fd=-1}],5,-1) = 1 ([fd=7,revents=POLLIN])
read(7,"\nParent process finished. Child "... ",4096) = 45
write(1,"\nParent process finished. Child "... ",45) = 45
shutdown(6,SHUT_WR) = 0
poll([fd=-1},{fd=7,events=POLLIN},{fd=-1},{fd=-1},{fd=-1}],5,-1) = 1 ([fd=7,revents=POLLIN])
read(7,"\33[?9001l\33[?1004l\n",4096) = 17
write(1,"\33[?9001l\33[?1004l\n",17) = 17
shutdown(6,SHUT_WR) = 0
poll([fd=-1},{fd=7,events=POLLIN},{fd=-1},{fd=-1},{fd=-1}],5,-1) = 1 ([fd=7,revents=POLLIN])
read(7,"",4096) = 0
shutdown(6,SHUT_WR) = 0
close(4) = 0
close(5) = 0
close(9) = 0
close(8) = 0
close(7) = 0
close(6) = 0
munmap(0x783f73036000,16384) = 0
ioctl(3,TCSETS,{c_iflag=ICRNL|IXON,c_oflag=NLO|CRO|TAB0|BS0|VT0|FF0|OPOST|ONLCR,c_cflag=0}) = 0
close(3) = 0
munmap(0x783f7303a000,8192) = 0
munmap(0x783f7303c000,4096) = 0
munmap(0x783f7303d000,4096) = 0
munmap(0x783f7303e000,4096) = 0
munmap(0x783f7303f000,4096) = 0
munmap(0x783f73040000,4096) = 0
munmap(0x783f73041000,4096) = 0
munmap(0x783f73042000,4096) = 0
munmap(0x783f73043000,4096) = 0
exit_group(0) = ?

```

```
+++ exited with 0 +++
```