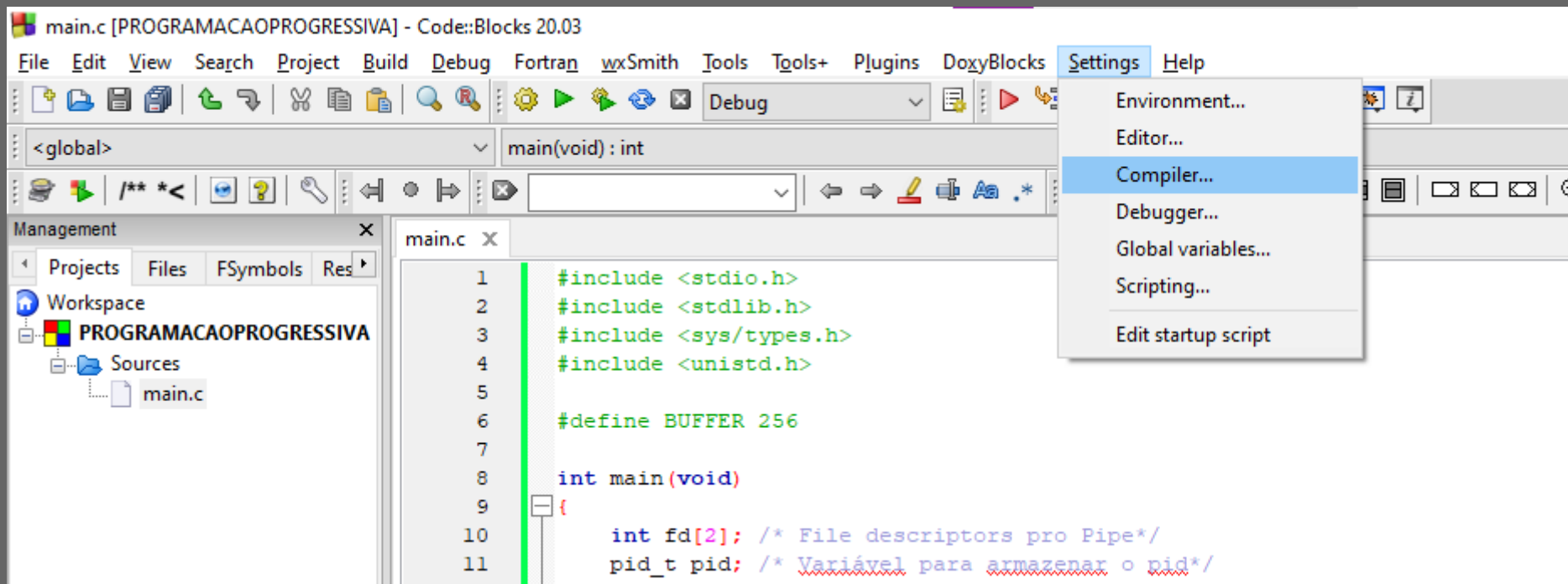


PIPES EM C
NANO
2023

PIPES NA TRADUÇÃO É CANO
OU SEJA UMA COMUNICAÇÃO ENTRE UMA PONTA E OUTRA.
ENTRE O PROCESSO PAI E SEU PROCESSO FILHO.

NO CODEBLOCKS



Global compiler settings



Global compiler settings



Profiler settings



Batch builds

Selected compiler

GNU GCC Compiler

Set as default

Copy

Rename

Delete

Reset defaults

Compiler settings

Linker settings

Search directories

Toolchain executables

Custom variables

Build

Compiler's installation directory

C:\Users\brito\Downloads\GNU_CYGWIN_WIN64\GNU_CYGWIN_WIN64



Auto-detect

NOTE: All programs must exist either in the "bin" sub-directory of this path, or in any of the "Additional

Program Files

Additional Paths

C compiler:

gcc.exe



C++ compiler:

g++.exe



Linker for dynamic libs:

g++.exe



Linker for static libs:

ar.exe



Debugger:

GDB/CDB debugger : Default



Resource compiler:

windres.exe



Make program:

mingw32-make.exe



OK

Cancel

PARA USAR A FUNÇÃO FORK()
NO WINDOWS TEMOS QUE BAIXAR O GNU_CYGWIN_WIN64
POIS ELA NÃO PODE SER USADA NO WINDOWS

Compiler settings

Global compiler settings



Global compiler settings

Selected compiler

GNU GCC Compiler

Set as default

Copy

Rename

Delete

Reset defaults

Compiler settings

Linker settings

Search directories

Toolchain executables

Custom variables

Build

Compiler's installation directory

C:\Users\brito\Downloads\GNU_CYGWIN_WIN64\GNU_CYGWIN_WIN64

Auto-detect

GNU_CYGWIN_WIN64

Arquivo

Início

Compartilhar

Exibir

Fixar no
Acesso rápido

Copiar

Colar

Recortar

Copiar caminho

Colar atalho

Mover
para

Copiar
para

Excluir

Renomear

Nova
pasta

Novo item
Fácil acesso

Propriedades

Abrir
Editar
Histórico

Selecionar tudo
Limpar seleção
Inverter seleção

Selecionar

Este Computador > Downloads > GNU_CYGWIN_WIN64

Pesquisar em GN...

OneDrive - Person

Este Computador

Área de Trabalho

Documentos

Downloads

Imagens

Músicas

Objetos 3D

Vídeos

Nome

Data de modificação

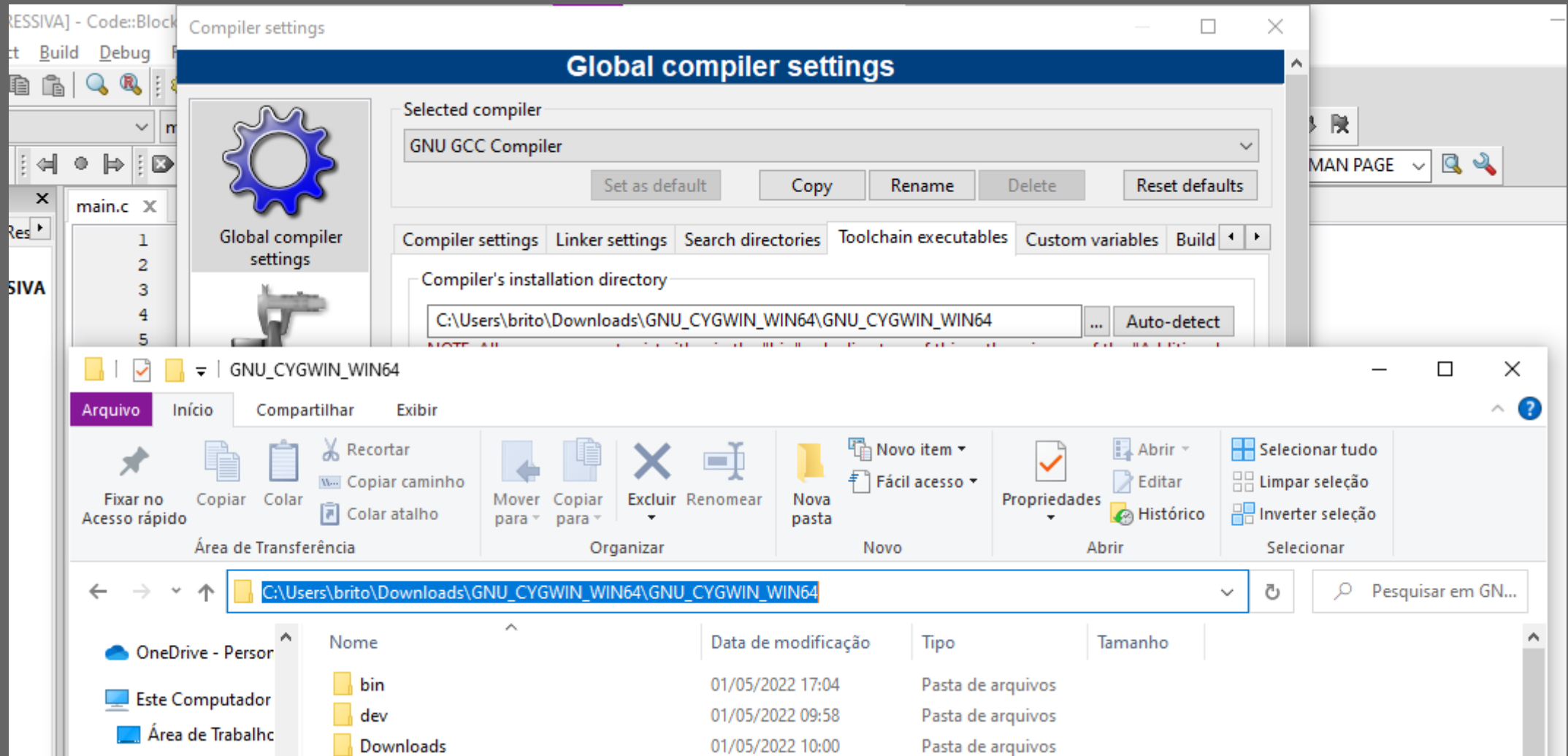
Tipo

Tamanho

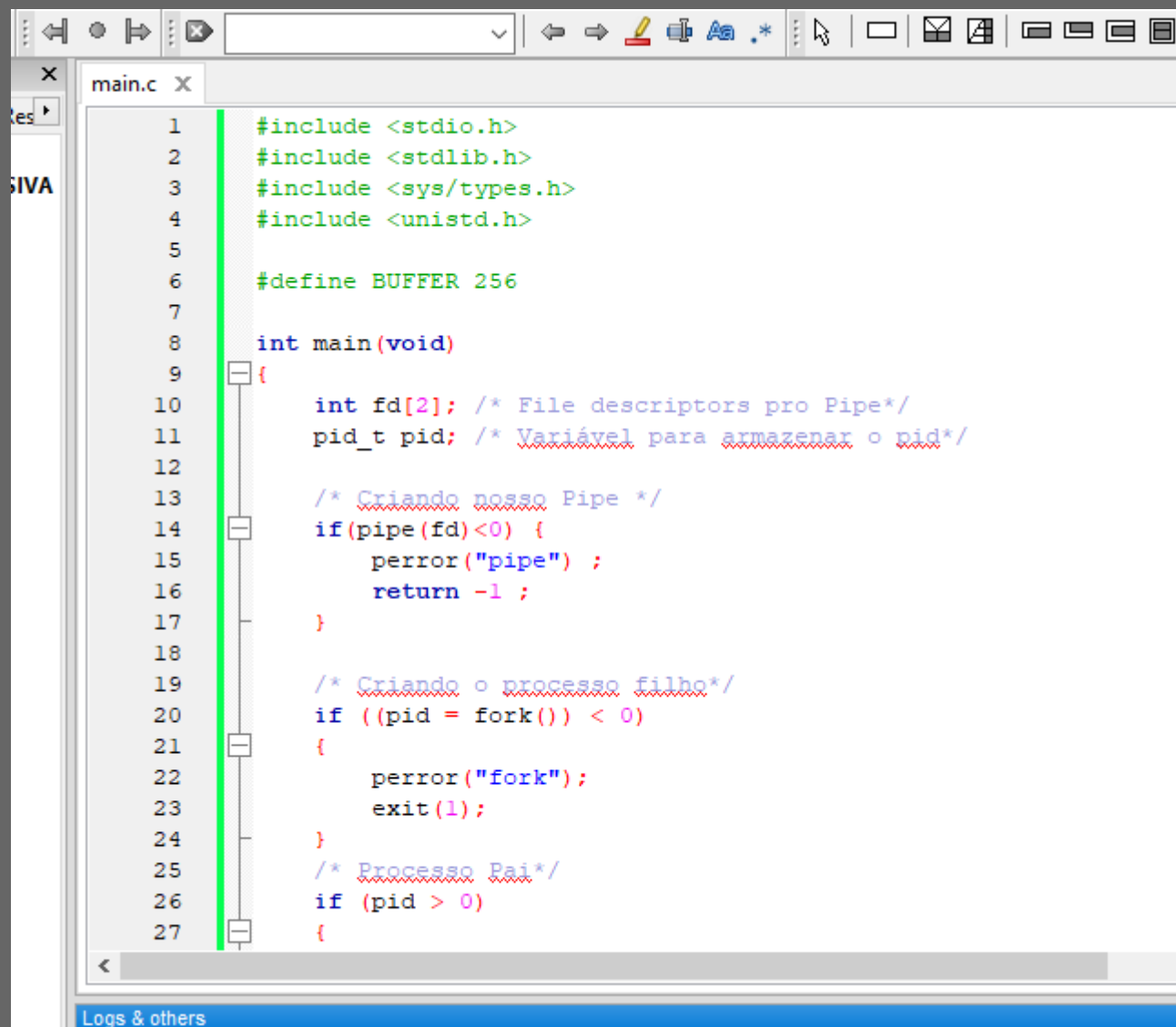
GNU_CYGWIN_WIN64

24/02/2023 11:34

Pasta de arquivos



ESCOLHI COMO CAMINHO ESSA PASTA QUE ESTA EM DOWNLOADS



The image shows a code editor window with a file named 'main.c'. The code is written in C and implements a simple process creation using a pipe. The editor has a standard toolbar at the top with icons for undo, redo, search, and other editing functions. The code is as follows:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <sys/types.h>
4  #include <unistd.h>
5
6  #define BUFFER 256
7
8  int main(void)
9  {
10     int fd[2]; /* File descriptors pro Pipe*/
11     pid_t pid; /* Variável para armazenar o pid*/
12
13     /* Criando nosso Pipe */
14     if(pipe(fd)<0) {
15         perror("pipe") ;
16         return -1 ;
17     }
18
19     /* Criando o processo filho*/
20     if ((pid = fork()) < 0)
21     {
22         perror("fork");
23         exit(1);
24     }
25     /* Processo Pai*/
26     if (pid > 0)
27     {
```

The code is partially visible, showing the beginning of the 'main' function and the first conditional check for the pipe creation. The editor interface includes a tab for 'main.c' and a status bar at the bottom that reads 'Logs & others'.


```
main.c X
25  /* Processo Pai*/
26  if (pid > 0)
27  {
28      /*No pai, vamos ESCREVER, então vamos fechar a LEITURA do Pipe neste lado*/
29      close(fd[0]);
30
31      char str[BUFFER] = "Apreendi a usar Pipes em C!";
32      printf("String enviada pelo pai no Pipe: '%s'", str);
33
34      /* Escrevendo a string no pipe */
35      write(fd[1], str, sizeof(str) + 1);
36      exit(0);
37  }
38  /* Processo Filho*/
39  else
40  {
41      char str_recebida[BUFFER];
42
43      /* No filho, vamos ler. Então vamos fechar a entrada de ESCRITA do pipe */
44      close(fd[1]);
45
46      /* Lendo o que foi escrito no pipe, e armazenando isso em 'str_recebida' */
47      read(fd[0], str_recebida, sizeof(str_recebida));
48
49      printf("String lida pelo filho no Pipe : '%s'\n\n", str_recebida);
50      exit(0);
51  }
```

main.c X

```
29     close(fd[0]);
30
31     char str[BUFFER] = "Aprendi a usar Pipes em C!";
32     printf("String enviada pelo pai no Pipe: '%s'", str);
33
34     /* Escrevendo a string no pipe */
35     write(fd[1], str, sizeof(str) + 1);
36     exit(0);
37 }
38 /* Processo Filho*/
39 else
40 {
41     char str_recebida[BUFFER];
42
43     /* No filho, vamos ler. Então vamos fechar a entrada de ESCRITA do pipe */
44     close(fd[1]);
45
46     /* Lendo o que foi escrito no pipe, e armazenando isso em 'str_recebida' */
47     read(fd[0], str_recebida, sizeof(str_recebida));
48
49     printf("String lida pelo filho no Pipe : '%s'\n\n", str_recebida);
50     exit(0);
51 }
52
53 return(0);
54 }
55
```

```

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

#define BUFFER 256

int main(void)
{
    int fd[2]; /* File descriptors pro Pipe*/
    pid_t pid; /* Variável para armazenar o pid*/

    /* Criando nosso Pipe */
    if(pipe(fd)<0) {
        perror("pipe") ;
        return -1 ;
    }

    /* Criando o processo filho*/
    if ((pid = fork()) < 0)
    {
        perror("fork");
        exit(1);
    }
    /* Processo Pai*/
    if (pid > 0)
    {
        /*No pai, vamos ESCREVER, então vamos fechar a LEITURA do Pipe neste lado*/
        close(fd[0]);

        char str[BUFFER] = "Aprendi a usar Pipes em C!";
        printf("String enviada pelo pai no Pipe: '%s'", str);

        /* Escrevendo a string no pipe */
        write(fd[1], str, sizeof(str) + 1);
        exit(0);
    }
    /* Processo Filho*/
    else
    {
        char str_recebida[BUFFER];

        /* No filho, vamos ler. Então vamos fechar a entrada de ESCRITA do pipe */
        close(fd[1]);

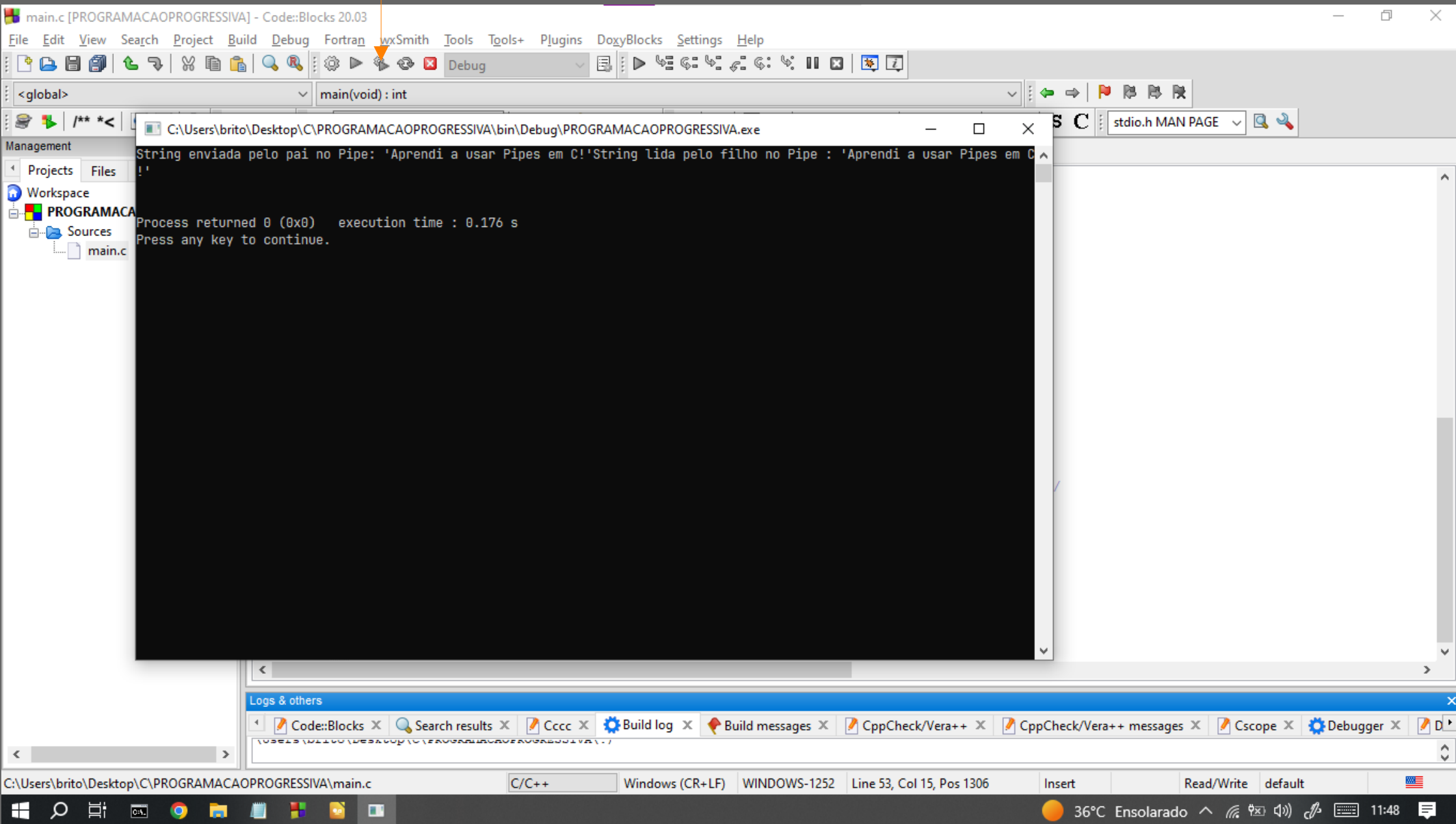
        /* Lendo o que foi escrito no pipe, e armazenando isso em 'str_recebida' */
        read(fd[0], str_recebida, sizeof(str_recebida));

        printf("String lida pelo filho no Pipe : '%s'\n\n", str_recebida);
        exit(0);
    }

    return(0);
}

```

CODE COMPLETO



COMPILANDO TEMOS A SAIDA ESPERADA

REFERÊNCIAS:

0.0 PIPE:

resolvi o erro do pipe usando o GNU_CYGWIN_WIN64
pois o pipe não é compatível com o windows. abaixo mostro como fiz.
no menu do codeblocks: Settings

```
|  
|--- Compiler  
|  
|--- Toolchain executables  
|  
|--- C:\Users\brito\Downloads\GNU_CYGWIN_WIN64\GNU_CYGWIN_WIN64
```

0.1 PASSOS:

Coloquei como caminho para o compilador a pasta do gnu_cgwin_win64 que se encontra em downloads
ou seja estou usando como compilador o cygwin

0.2 achei isso nesse link de tutorial:

<https://stackoverflow.com/questions/8819673/undefined-reference-to-fork-in-codeblocks-editor-in-windows-os>

SENDO ASSIM:

Avanço nos estudos e fico glorificando a deus.

Como estudo gcc

Ai esta os logs da saida

Logs & others

Code::Blocks X Search results X Cccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X D

Target is up to date.

Nothing to be done (all items are up-to-date).

----- Run: Debug in PROGRAMACAOPROGRESSIVA (compiler: GNU GCC Compiler)-----

Checking for existence: C:\Users\brito\Desktop\C\PROGRAMACAOPROGRESSIVA\bin\Debug\PROGRAMACAOPROGRESSIVA.exe

Set variable: PATH=.;C:\Users\brito\Downloads\GNU_CYGWIN_WIN64\GNU_CYGWIN_WIN64\bin;C:\Users\brito\Downloads\GNU_CYGWIN_WIN64\GNU_CYGWIN_WIN64;C:\Program Files\CodeBlocks\MinGW\bin;C:\cygwin64\bin;C:\cygwin64;C:\Program Files\CodeBlocks\MinGW;C:\Program Files\Java\jdk-17\bin;C:\Program Files\Microsoft\jdk-11.0.12.7-hotspot\bin;C:\Users\brito\AppData\Local\Programs\Python\Python39\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python39;C:\Program Files (x86)\VMware\VMware Player\bin;C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Windows\System32;C:\Windows;C:\Windows\System32\wbem;C:\Windows\System32\WindowsPowerShell\v1.0;C:\Windows\System32\OpenSSH;C:\Program Files\CodeBlocks\MinGW\lib\gcc\x86_64-w64-mingw32\lib;C:\msys64\mingw64\bin;C:\MinGW\bin;C:\Program Files\Microsoft SQL Server\130\Tools\Binn;C:\Program Files\Java\javafx-sdk-16\bin;C:\Program Files\Java\jdk-11.0.11\bin;C:\Program Files\Java\jre1.8.0_291\bin;C:\xampp\php;C:\Program Files\MySQL\MySQL Server 8.0\bin;C:\Program Files\dotnet;C:\Program Files\CMake\bin;C:\wxwidgets\lib\vc142_x64_dll;C:\sqlite3;C:\SQLiteStudio;C:\Microsoft VS Code\bin;C:\opencv\build\x64\vc15\bin;C:\Program Files\TortoiseSVN\bin;C:\Windows\Microsoft.NET\Framework64\v4.0.30319;C:\Users\brito\Documents\phaser\scripts\tsgen\bin;C:\Program Files\JetBrains\PyCharm Community Edition 2021.3\bin;C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.1.3\bin;C:\Program Files\JetBrains;C:\Program Files\JetBrains\PyCharm Community Edition 2021.3;C:\OpenCV2\opencv\build\bin;C:\opencv\build\bin;C:\Program Files (x86)\Microchip\MPLAB C32 Suite\bin;C:\Program Files\MongoDB\Server\5.0\bin;C:\Users\brito\AppData\Local\atom\app-1.58.0;C:\opencv\build\x64\vc14\bin;C:\TDM-GCC-32\bin;C:\Windows\SysWOW64\freeglut.dll;C:\GLUT\bin;C:\msys64\mingw64\include\gtk-3.0;C:\msys64\mingw64\include\gtk-4.0;C:\gradle-7.4.1\bin;C:\Program Files\netcat-1.12;C:\apache-maven-3.8.6\bin;C:\gdb\bin;C:\Users\brito\AppData\Roaming\nvm;C:\Program Files\nodejs;C:\Neovim\bin;C:\Vim\vim90;C:\Program Files\Git\cmd;C:\MinGW\32bits\MinGW-4.8.1\mingw32\bin;C:\ProgramData\chocolatey\bin;C:\Users\brito\scoop\shims;C:\Users\brito\AppData\Local\Programs\Python\Python311\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python311;C:\Users\brito\AppData\Local\Programs\Python\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\WindowsApps;C:\MinGW\64bits\MinGW-4.8.1\mingw64\bin;C:\Users\brito\dotnet\tools;C:\Users\brito\AppData\Local\atom\bin;C:\Users\brito\AppData\Local\Programs\oh-my-posh\bin;C:\LLVM\64bits\LLVM-15.0.7\bin;C:\LLVM\32bits\LLVM-3.5.0\bin;C:\LLVM\64bits\LLVM-3.7.0\bin;C:\LLVM\32bits\LLVM-3.4\bin;C:\LLVM\32bits\LLVM-3.7.0\bin;C:\LLVM\32bits\LLVM-5.0.0\bin;C:\LLVM\32bits\LLVM-9.0.0\bin;C:\LLVM\32bits\LLVM-11.0.0\bin;C:\LLVM\32bits\LLVM-14.0.6\bin;C:\LLVM\32bits\LLVM-15.0.7\bin;C:\LLVM\64bits\LLVM-5.0.0\bin;C:\LLVM\64bits\LLVM-9.0.0\bin;C:\LLVM\64bits\LLVM-11.0.0\bin;C:\LLVM\64bits\LLVM-14.0.6\bin;C:\MinGW\64bits\MinGW-4.8.5\mingw64\bin;C:\MinGW\64bits\MinGW-7.2.0\mingw64\bin;C:\MinGW\32bits\MinGW-4.9.4\mingw32\bin;C:\MinGW\32bits\MinGW-5.4.0\mingw32\bin;C:\MinGW\32bits\MinGW-8.1.0\mingw32\bin

Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\PROGRAMACAOPROGRESSIVA\bin\Debug\PROGRAMACAOPROGRESSIVA.exe" (in C:\Users\brito\Desktop\C\PROGRAMACAOPROGRESSIVA\.)

29
30

`close(fd[0]);`

Logs & others

Code::Blocks X

Search results X

Cccc X

Build log X

Build messages X

CppCheck/Vera++ X

CppCheck/Ve

File	Line	Message
		=== Build: Debug in PROGRAMACAOPROGRESSIVA (compiler: GNU GCC Compiler) ===
		=== Run: Debug in PROGRAMACAOPROGRESSIVA (compiler: GNU GCC Compiler) ===

Senso assim:

0.0 fica comprovado que o gcc deve ser estudado

0.1 o windows é hostil com outras ferramentas

0.2 existe mais de um jeito de fazer as coisas

0.3 a compatibilidade sempre vai ser um problema?

0.4 sempre devemos tentar resolver esses problemas

0.5 continuar estudando pipe

5.0 objetivo de estudar pipe é criar uma ferramenta de pesquisas autonoma que busque no google e tenha a capacidade de interagir com ele. Ex: enviando um e-mail.

5.1 Obs. como aquela ferramenta que usei no terminal com python que faz um brute force. E como a ferramenta do git que baixa os repositórios.


```
----- Build: Debug in PROGRAMACAOPROGRESSIVA (compiler: GNU GCC Compiler)-----  
gcc.exe -Wall -g -c C:\Users\brito\Desktop\C\PROGRAMACAOPROGRESSIVA\main.c -o obj\Debug\main.o  
gcc.exe -o bin\Debug\PROGRAMACAOPROGRESSIVA.exe obj\Debug\main.o  
Output file is bin\Debug\PROGRAMACAOPROGRESSIVA.exe with size 69.39 KB  
Process terminated with status 0 (0 minute(s), 1 second(s))  
0 error(s), 0 warning(s) (0 minute(s), 1 second(s))  
  
----- Run: Debug in PROGRAMACAOPROGRESSIVA (compiler: GNU GCC Compiler)-----
```