# Terminal:

cd – change directory

ls – list

CNTR + L – clean the terminal

gcc -o \*(Inserir nome do ficheiro compilado) \*(inserir nome do ficheiro VIM) - compila

./ \*(Inserir nome do ficheiro compilado) – corre o compilado

# Código:

**WRITE:**

#include <unistd.h> Uma imagem com texto, captura de ecrã, file, Tipo de letra

Descrição gerada automaticamente

EX: write(1,”fizzbuzz”,8); - escreve a palavra “fizzbuzz” no terminal 8 bytes

write(1,&C,1) – pointer para a variável char C

Para escrever um Integer:

1 maneira

void ft\_write\_number(int number){

char str[10] = "0123456789";

if (number > 9)

ft\_write\_number(number / 10);

write (1, &str[number % 10], 1);}

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

2 maneira

Int a = ‘0’;

write(1,&a,1);

usar depois sempre as ‘’ para por números e comparar com o Int a

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, software, Tipo de letra

Descrição gerada automaticamente

**Data Types**

Int a = ‘0’;

Int b = ‘0’;

A = B + 1; reparar em como não se usa as aspas para adicionar 1

While(a < ‘10’) mas nas comparações usa-se aspas

**Recursive call**

Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra, documento

Descrição gerada automaticamente

Exemplo:

Nb=31456

/10 até nb = 3;

Volta ao estado anterior(nb = 31) e executa a segunda linha de codigo nb % 10, que dá 1.

Faz isto ate tar todos os números impessos

Para guardar um conjunto de caracteres

Ola[50] = “adasdasda”; - se não definir o tamamho do array o compilador faz automaticamente

&ola[0] – manda o endereço do caracter 0

Str[i] != ‘\0’ – o \0 representao final dos char

Depois corre todo o conjunto de endereços onde está o “ola[50]” até encontrar o final

Uma imagem com texto, captura de ecrã, software

Descrição gerada automaticamente