Rush00

Introducción

Este documento sirve para explicar el código necesario para hacer los diferentes ejercicios del Rush00.

Por favor, hay que evitar copiar el código. Os haceis un flaco favor, debéis de entender como se hacen por vosotros mismos y la lógica detrás de ello, no solo os ayudará en futuras ocasiones, si no que además, hará que obtengáis una buena puntuación a la hora de defenderlo.

Enunciado

main.c

La base del Rush00 que todos los grupos deben hacer es preparar los archivos *main.c* y *ft_putchar.c*.

En el archivo main.c es donde llamaremos a nuestra función rush y le daremos valor a sus variables, ya que es lo que ejecutará. Este es el código:

```
void rush(int x, int y);

int main(void)
{
    rush(4, 4);
    return (0);
}
```

Explicación del código

Para entender el código, voy a desglosar y explicar cada parte del mismo, tratando de que incluso con poca experiencia programando, podáis entenderlo.

1. Declaración de la función *rush*

```
void rush(int x, int y);
```

- **void:** Indica que la función rush no devolverá ningun valor.
- **rush:** Este es el nombre de la función a la que está llamando.
- **(int x, int y):** La función rush tomará dos parámetros de números enteros (x e y).

2. Función **main**

```
int main(void)
{
// Código
}
```

- **int:** Indica que la función *main* devolverá un valor entero.
- **main:** Esta es la función principal de todo programa en C. El código una vez compilado ejecutará esta función, y lo que haya dentro de ella.
- **(void):** Indica que la función *main* no hace uso de ningún parámetro/variable.

3. Llamada a la función *rush*:

```
rush(4, 4);
```

- Esta línea llama a la función rush con los parámetros/variables 4 y 4. Esto significa en el contexto de nuestra función, que trabajará con *x = 4*, *y = 4*, es decir, que hará aparecer en pantalla una figura de 4 x 4.

4. Retorno de la función *main*:

```
return (0);
```

- **return (0):** Indica que el programa ejecuta todo el código correctamente. Se devuelve a la función *main* el valor *0*, lo que en *C* significa éxito.

Función del archivo *main.c*

El archivo *main.c* sirve como el punto de inicio del programa. Define la función main que llama a la función *rush*, añadiendo los valores deseados para sus parametros (*x* e *y*). Aquí es donde configuramos lo que nuestro programa hará al ejecutarse.

ft_putchar.c

En este archivo tendremos la función que permitirá mostrar por pantalla los caracteres que deseemos, el código es el siguiente:

```
#include <unistd.h>

void ft_putchar(char c)
{
    write(1, &c, 1);
}
```

Explicación del código

1. Incluir la biblioteca *unistd.h*

```
#include <unistd.h>
```

- Esta línea incluye la biblioteca *unistd.h*, que contiene la definición de la función *write*. Esta función se utiliza para la salida de datos en *Linux*.

2. Definición de la función *ft_putchar*

```
void ft_putchar(char c)
{
// Código
}
```

- **void**: Indica que la función ft_putchar no devolverá ningún valor.
- **ft_putchar**: Nombre de la función.
- **(char c)**: La función tomará un único parámetro/variable de tipo caracter.

3. Llamada a la función *write*

```
write(1, &c, 1);
```

- **write**: Esta es una llamada al sistema que envía datos para ser mostrados.
- **1**: Especifica que estamos escribiendo a la salida estandar (pantalla).
- **&c**: Es la dirección del carácter que queremos escribir, en este caso es la variable c que hemos definido al crear la función.
- **1**: Indica la cantidad de bytes que va a escribir, en este caso, al ser un solo carácter, 1.

Función del archivo ft_putchar.c

El archivo ft_putchar.c define una función que nos permite mostrar caracteres individuales en la pantalla. Esta función es fundamental para crear cualquier salida visual en nuestro programa, como dibujar formas o imprimir texto en pantalla.

Ejercicio 0 (Rush00)

Este es el archivo rush.c para el ejercicio 0 (Rush00), el cual es esencial para el ejercicio. El código es el siguiente:

```
void ft_putchar(char c);

void draw(int row, int column, int x, int y)
{
    if ((row == 0 && column == 0)
        || (row == 0 && column == x - 1)
        || (row == y - 1 && column == 0)
        || (row == y - 1 && column == x - 1))
    {
        ft_putchar('o');
    }
    else if (row == 0 || row == y - 1)
    {
        ft_putchar('-');
    }
    else if (column == 0 || column == x - 1)
    {
        ft_putchar('|');
    }
    else
    {
        ft_putchar(' ');
    }
}

void rush(int x, int y)
{
    int row;
    int column;

    row = 0;
    while (row < y)
    {
        column = 0;
        while (column < x)
        {
            draw(row, column, x, y);
            column++;
        }
        ft_putchar('\n');
        row++;
    }
}
```

Explicación del código

Función *draw*

Al comenzar el archivo, declaramos al principio la función *ft_putchar* del otro archivo que hemos creado anteriormente en el enunciado, no se le añade codigo ya que su definición real se encuentra en *ft_putchar.c*

```
void ft_putchar(char c);
```

Función *draw*

Esta función se encarga de determinar con condicionales (*if*) que caracter se debe imprimir en cada posición de la figura, de acuerdo con las coordenadas actuales ('fila', 'columna') y las dimensiones ('ancho', 'alto').

```
void draw(int row, int column, int x, int y)
{
// Esquina superior izquierda, superior derecha, inferior izquierda e inferior derecha
    if ((row == 0 && column == 0)
        || (row == 0 && column == x - 1)
        || (row == y - 1 && column == 0)
        || (row == y - 1 && column == x - 1))
    {
        ft_putchar('o');
    }
    // Borde superior e inferior
    else if (row == 0 || row == y - 1)
    {
        ft_putchar('-');
    }
    // Borde izquierdo y derecho
    else if (column == 0 || column == x - 1)
    {
        ft_putchar('|');
    }
    // Espacio interior
    else
    {
        ft_putchar(' ');
    }
}
```

- **Esquinas**: Imprime en pantalla 'o' en las cuatro esquinas de la figura.
- **Bordes horizontales**: Imprime en pantalla '-' en la primera y última fila, exceptuando las esquinas.
- **Bordes verticales**: Imprime '|' en la primera y ultima columna, exceptuando las esquinas, ya que en este punto de la condicional ya se han rellenado, por lo que no hay que preocuparse.
- **Interior**: Imprime un espacio en blanco para todas las otras posiciones, por lo general, el centro de la figura, el cual debe ser hueco.

Función *rush*

Esta función es la encargada de iterar a través de cada posición en la figura y llamar a la función draw para determinar que carácter imprimir según la posición, todo esto lo hace a través de un bucle anidado.

```
void rush(int x, int y)
{
    int row;
    int column;

    row = 0;
    while (row < y)
    {
        column = 0;
        while (column < x)
        {
            draw(row, column, x, y);
            column++;
        }
        ft_putchar('\n');
        row++;
    }
}
```

- **Iteración por filas y columnas**: Utiliza un bucle anidado, conformado por dos bucles 'while' para recorrer todas las posiciones de la figura.
- **Llamada a 'draw'**: En cada posición '(fila, columna)', llama a draw para determinar el caracter a imprimir por pantalla.
- **Salto de línea**: Después de terminar una fila, imprime un salto de línea ('\n').

Compilación

Finalmente, para compilar todos los archivos ('*ft_putchar.c*', '*main.c*' y '*rush0x.c*'), se utiliza el siguiente comando en la terminal:

```
cc -Wall -Wextra -Werror ft_putchar.c main.c rush0x.c -o rush0x
```

- **cc**: Es el comando para compilar en C.
- **ft_putchar.c main.c rush0x.c**: Lista de archivos que se van a compilar.
- **-o rush0x**: Indica el nombre del archivo que va a crearse al compilar.

Vocabulario

- **Iterar**: Significa repetir, en este caso se usa para hacer mención a los bucles.

Hecho por Marcos03BR