GTK Installation:

Download Windows Terminal from Microsoft store(or use ubuntu OS terminal i.e WSL)

```
sudo apt-get upgrade
sudo apt-get update
sudo apt-get install build-essential
sudo apt-get install libgtk2.0-doc devhelp
sudo apt-get install glade
```

*if it didn't work Follow this video upto (2:09 min)
https://www.youtube.com/watch?v=Txo1OFViWpU

```
(glade:391): Gtk-WARNING **: 02:58:09.056: Error loading theme icon 'dialog-warning' for
stock: Icon 'dialog-warning' not present in theme Adwaita
--> sudo apt install adwaita-icon-theme-full
```

GTK+ 2.0 Tutorial

GUI USING POINTERS (https://www.gtk.org/)

GTK(GNU Image Manipulation Program Toolkit)

GNOME Builder :

GNOME Builder is a general-purpose integrated development environment (IDE) for the GNOME platform, primarily designed **to aid in writing GNOME-based applications**.

1.What is GTK?

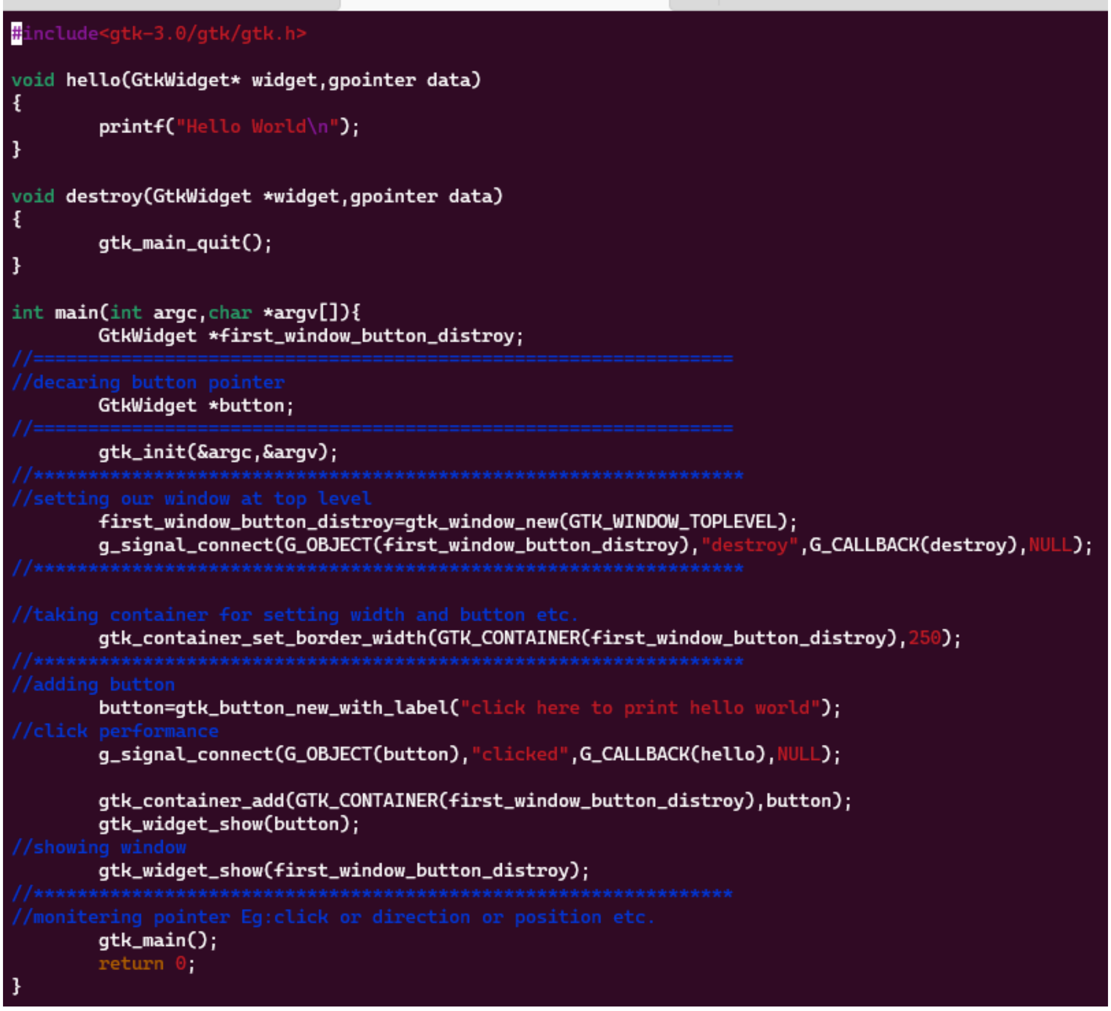
GTK is a free and **open-source cross-platform** widget toolkit for creating graphical user interfaces. It is licensed under the terms of the GNU Lesser General Public License, allowing both free and proprietary software to use it. It is one of the most popular toolkits for the Wayland and X11 windowing systems.

2.Why we used Glade for GUI?

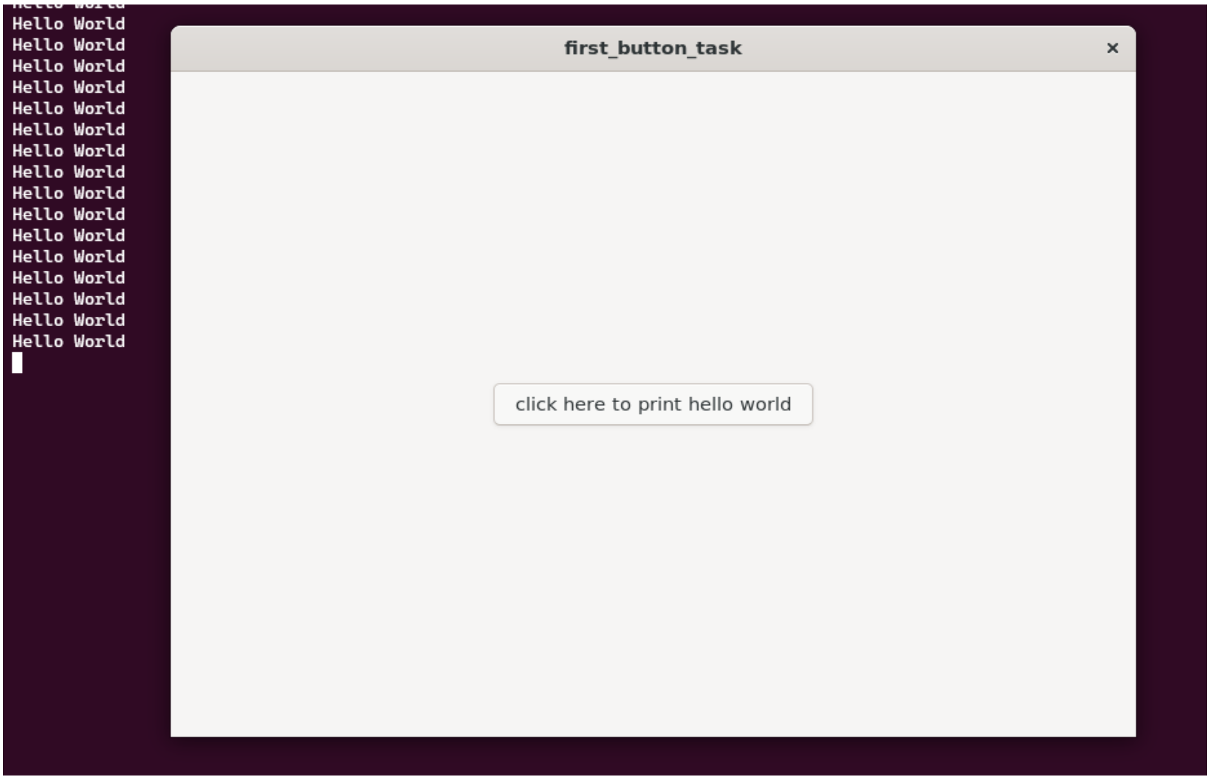
You can use Glade to place widgets in a GUI. Glade **allows you to modify the layout and properties of these widgets**. You can also use Glade to add connections between widgets and application source code. The user interfaces designed in Glade are stored in an XML format, enabling easy integration with external tools.

It is cumbersome to write and handle every widget on screen.

Following Code is for to print Hello World on Terminal without using glade.

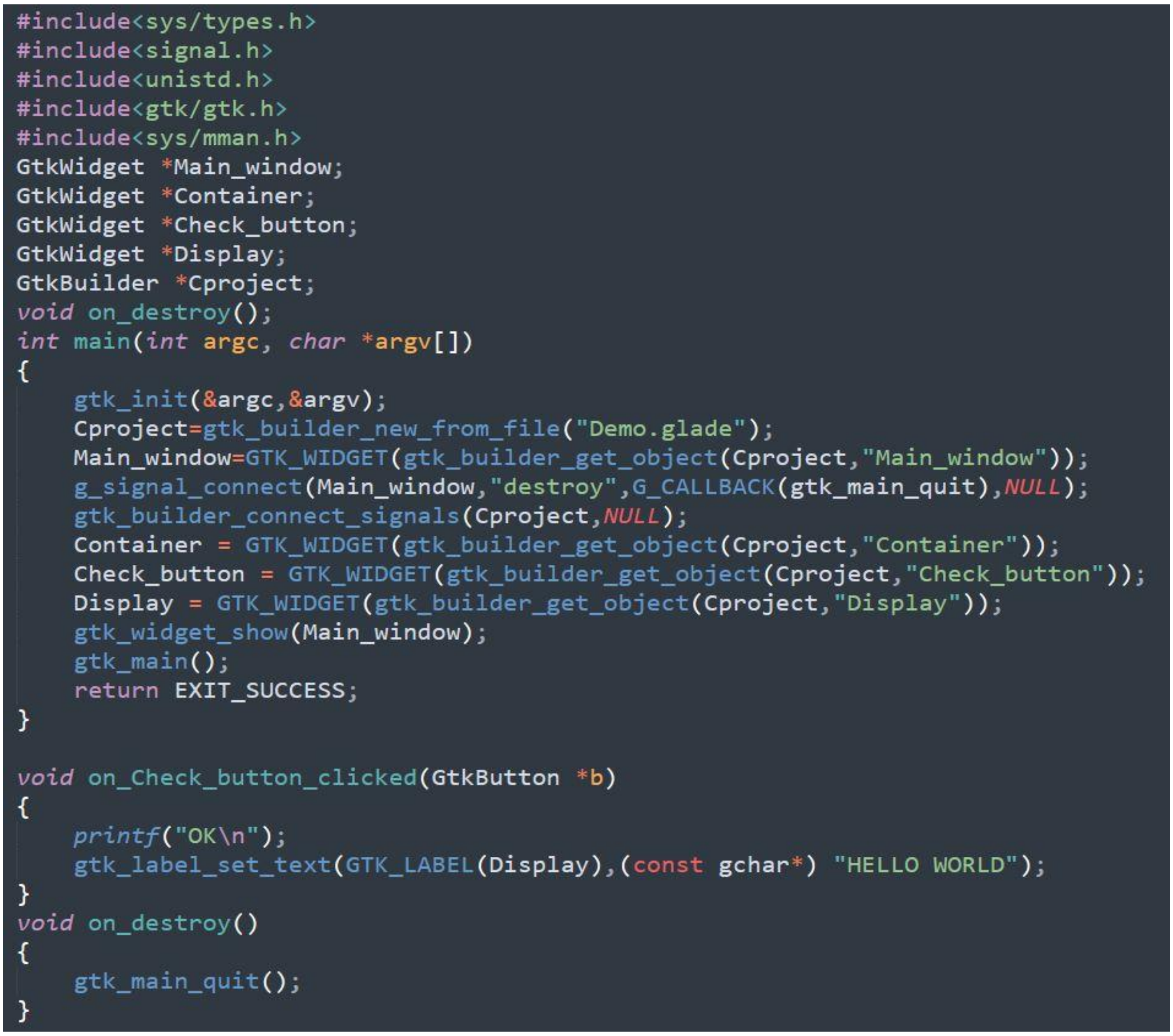


```c
#include<gtk-3.0/gtk/gtk.h>

void hello(GtkWidget* widget,gpointer data)
{
        printf("Hello World\n");
}

void destroy(GtkWidget *widget,gpointer data)
{
        gtk_main_quit();
}

int main(int argc,char *argv[]){
        GtkWidget *first_window_button_distroy;
//================================================================
//decaring button pointer
        GtkWidget *button;
//================================================================
        gtk_init(&argc,&argv);
//*****************************************************************
//setting our window at top level
        first_window_button_distroy=gtk_window_new(GTK_WINDOW_TOPLEVEL);
        g_signal_connect(G_OBJECT(first_window_button_distroy),"destroy",G_CALLBACK(destroy),NULL);
//*****************************************************************

//taking container for setting width and button etc.
        gtk_container_set_border_width(GTK_CONTAINER(first_window_button_distroy),250);
//*****************************************************************
//adding button
        button=gtk_button_new_with_label("click here to print hello world");
//click performance
        g_signal_connect(G_OBJECT(button),"clicked",G_CALLBACK(hello),NULL);

        gtk_container_add(GTK_CONTAINER(first_window_button_distroy),button);
        gtk_widget_show(button);
//showing window
        gtk_widget_show(first_window_button_distroy);
//****************************************************************
//monitering pointer Eg:click or direction or position etc.
        gtk_main();
        return 0;
}
```



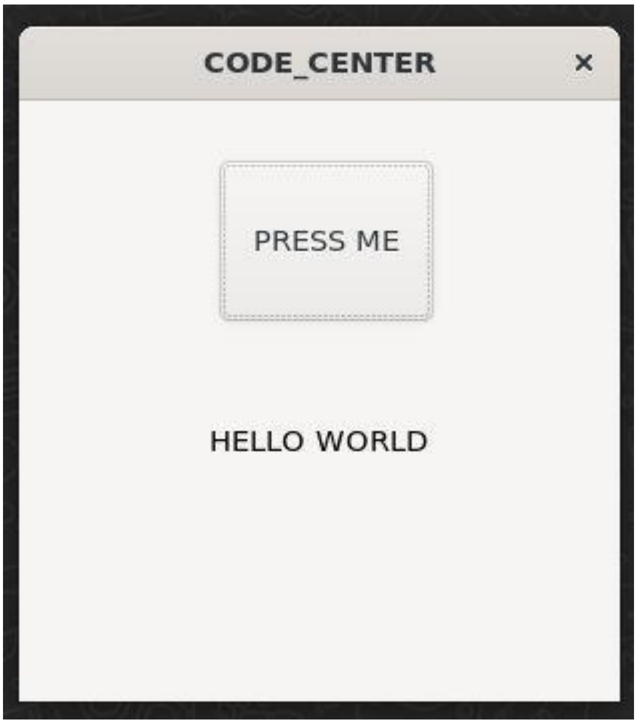
```
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
```

Following Code is for to print Hello World on GUI with using glade:-



```
#include<sys/types.h>
#include<signal.h>
#include<unistd.h>
#include<gtk/gtk.h>
#include<sys/mman.h>
GtkWidget *Main_window;
GtkWidget *Container;
GtkWidget *Check_button;
GtkWidget *Display;
GtkBuilder *Cproject;
void on_destroy();
int main(int argc, char *argv[])
{
    gtk_init(&argc,&argv);
    Cproject=gtk_builder_new_from_file("Demo.glade");
    Main_window=GTK_WIDGET(gtk_builder_get_object(Cproject,"Main_window"));
    g_signal_connect(Main_window,"destroy",G_CALLBACK(gtk_main_quit),NULL);
    gtk_builder_connect_signals(Cproject,NULL);
    Container = GTK_WIDGET(gtk_builder_get_object(Cproject,"Container"));
    Check_button = GTK_WIDGET(gtk_builder_get_object(Cproject,"Check_button"));
    Display = GTK_WIDGET(gtk_builder_get_object(Cproject,"Display"));
    gtk_widget_show(Main_window);
    gtk_main();
    return EXIT_SUCCESS;
}

void on_Check_button_clicked(GtkButton *b)
{
    printf("OK\n");
    gtk_label_set_text(GTK_LABEL(Display),(const gchar*) "HELLO WORLD");
}
void on_destroy()
{
    gtk_main_quit();
}
```

Output Window :-



- Every entity present on GUI is widget.

- Every widget is represented by pointer will have their own on click event or function.

- Following function will return a value and according to our business logic we will get our desired output.

- Eg.

  ```
  GtkWidget *strshowcode;                          (A Pointer)
  void on_strshowcode_clicked(GtkButton *b)        (Function Call)
  ```

- GtkWidget is a generic pointer .

- For every widget we create Their own pointer to point at them.

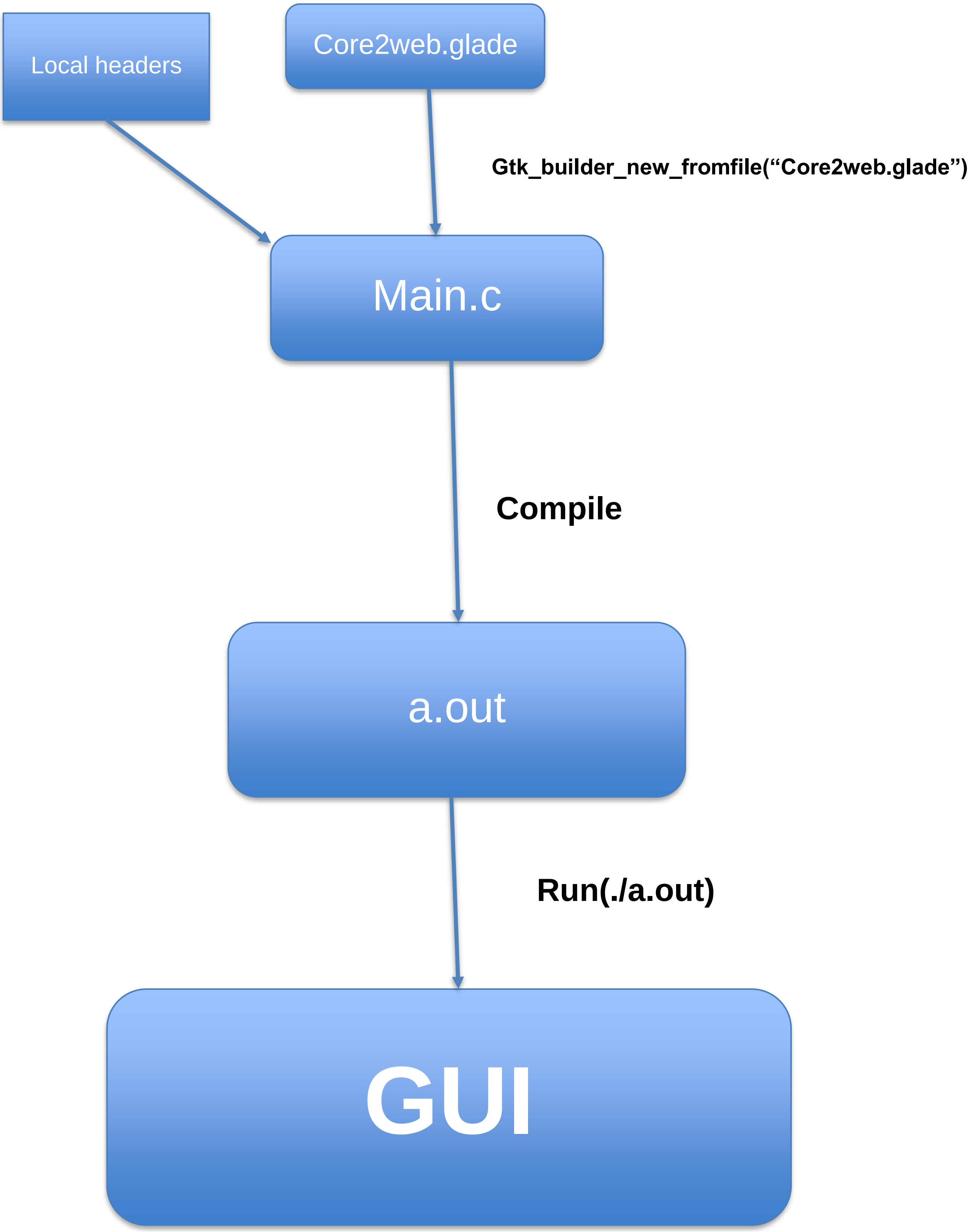
- For every pointer it is necessary to built a builder for that particular widget.
  Eg.

  ```
  Container = GTK_WIDGET(gtk_builder_get_object(Cproject, "Container"));
  ```

  Here Cproject is our .c file name and Container is a pointer to a widget.

- It’s like reference and object relation.

- Those pointers will handle those runtime onclick events.



Local headers
Core2web.glade
Gtk_builder_new_fromfile("Core2web.glade")
Main.c
Compile
a.out
Run(./a.out)
GUI