VIII Jornadas de Software Libre

Metiendo mano en un proyecto de SL: modificando lxrandr

Joaquín Bogado – Andrea Gómez Del Mónaco

Entorno de escritorio en Sistemas GNU/Linux

Es un conjunto de aplicaciones que permiten una interacción dinámica, amigable y eficiente entre el usuario y el sistema operativo.

Está compuesto por:

- Escritorio
- Paneles
- Ventanas
- Áreas de trabajo

Entornos de escritorio en Sistemas GNU/Linux

LXDE

"Lightweight X11 Desktop Environment"

Entorno de Escritorio X11 Liviano

- Bajo consumo de CPU y memoria RAM
- Diseñado para máquinas con hardware limitado
  - Computadoras antiguas /MIDs)
  - Dispositivos móviles
  - Netbooks (ej: Proyecto edubooks)

Su código fuente está disponible bajo los término de la Licencia Pública General (GPL)

lxrandr

La aplicación es una interfaz gráfica para el comando ***xrandr***, un programa de línea de comandos para administrar la resolución de la pantalla y de la tasa de refresco de el/los monitor/es.

lxrandr

Descripción del paquete

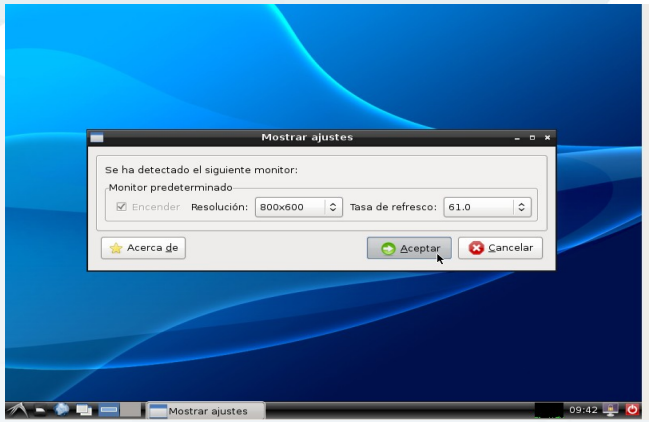
- Código fuente escrito en C
- Librería utilizada para la interfaz gráfica: GTK
- Versión: 0.1.1
- Maintainer: Andrew Lee

“Description: simple monitor config tool for LXDE
This is a very basic monitor config tool utilizing the X extension called RandR. It can let you change the screen resolution on the fly. Besides, when you run lxrandr with external monitor connected, its GUI will change, and show you some quick options to get your projector or external monitor working correctly”.

lxrandr

Inconvenientes:

Los cambios no se almacenan permanentemente.



Los usuarios deben configurar las opciones de administración de monitor cada vez que enciende el equipo o reinicia su sesión.

lxrandr

Como los cambios no son almacenados, si el usuario hace logout de su sesión, el sistema vuelve a setear las opciones de configuración del monitor del archivo ubicado en

/etc/X11/xor.conf

```
Section "Screen"
    Identifier     "Screen0"
    Device         "Device0"
    Monitor        "Monitor0"
    DefaultDepth    24
    Option         "TwinView" "0"
    Option         "metamodes" "1024x768 +0+0"
    SubSection     "Display"
        Depth       24
    EndSubSection
EndSection
```

Inconvenientes

- Todos los usuarios deben utilizar las mismas configuraciones de pantalla
- Sólo el usuario **root** tiene privilegio para modificar éste archivo

lxrandr

Funcionamiento

Estructuras de lxrandr

```
typedef struct _Monitor
{
    char* name;
    GSList* mode_lines;
    short active_mode;
    short active_rate;
    short pref_mode;
    short pref_rate;

    GtkCheckButton* enable;
    GtkComboBox* res_combo;
    GtkComboBox* rate_combo;
}Monitor;
```

```
static GSList* monitors = NULL;
static Monitor* LVDS = NULL;

    static GtkWidget* dlg = NULL;
```

static void get_xrandr_info()

```
static gboolean get_xrandr_info()
{
    GRegex* regex;
    GMatchInfo* match;
    int status;
    char* output = NULL;
    char* ori_locale;

    ori_locale = g_strdup( setlocale(LC_ALL, "") );

    // set locale to "C" temporarily to guarantee English output of xrandr
    setlocale(LC_ALL, "C");

    if( ! g_spawn_command_line_sync( "xrandr", &output, NULL, &status, NULL ) || status )
    {
        g_free( output );
        setlocale( LC_ALL, ori_locale );
        g_free( ori_locale );
        return FALSE;
    }

   regex = g_regex_new( "([a-zA-Z]+[-0-9]*) +connected .*((\n +[0-9]+x[0-9]+[^\n]+)+)",
                        0, 0, NULL );
    if( g_regex_match( regex, output, 0, &match ) )
    {
        Do {
                            [..]
```

A través de la función **`get_xrandr_info()`**, se solicitan los modos de resoluciones como de tasas de refresco soportados por la placa de video y el/los monitor/es

Mian de lxrand

```
int main(int argc, char** argv)
{
    GtkWidget *notebook, *vbox, *frame, *label, *hbox, *check, *btn;
    GSList* l;

        [...]


      gtk_combo_box_append_text( m->res_combo, _("Auto") );
      for( mode_line = m->mode_lines; mode_line; mode_line = mode_line->next )
      {
         char** strv = (char**)mode_line->data;
         gtk_combo_box_append_text( m->res_combo, strv[0] );
      }

      gtk_combo_box_set_active( m->res_combo, m->active_mode + 1 );
      gtk_combo_box_set_active( m->rate_combo, m->active_rate + 1 );
    }

    gtk_widget_show_all( dlg );

    if( gtk_dialog_run( (GtkDialog*)dlg ) == GTK_RESPONSE_OK )
      set_xrandr_info();

    gtk_widget_destroy( dlg );

        return 0;
}
```

Una vez que el usuario presiona el botón “Aceptar” se hace una llamada a la función ***set_xrandr_info()***

static void set_xrandr_info()

```
{
    GSList* l;
    GString *cmd = g_string_sized_new( 1024 );

    g_string_assign( cmd, "xrandr" );

    for( l = monitors; l; l = l->next )
    {
        Monitor* m = (Monitor*)l->data;
        g_string_append( cmd, " --output " );
        g_string_append( cmd, m->name );
        g_string_append_c( cmd, ' ' );

        // if the monitor is turned on
        if( gtk_toggle_button_get_active( GTK_TOGGLE_BUTTON(m->enable) ) )
        {
            int sel_res = gtk_combo_box_get_active( m->res_combo );
            int sel_rate = gtk_combo_box_get_active( m->rate_combo );

            if( sel_res < 1 ) // auto resolution
            {
                g_string_append( cmd, "--auto" );
            }
            else
                    [..]
}
```

Encargada de invocar al comando **xrandr** con los parámetros determinados por los valores de los widgets de la interfaz gráfica que el usuario ha seleccionado.

$ xrandr --output LVDS --mode 1280x800 --output VGA-0 --mode 1024x768 --right-of LVDS

Objetivos

- Modificar la función **set_xrandr_info()** de forma tal que guarde los estados de los widget de la herramienta en el momento que se presiona el botón **aceptar**.

- Mantener la información de los argumentos del comando **xrandr** en un script que se ejecute cada inicio de sesión.

- Guardar el script en la carpeta personal de cada usuario. De esta manera:
  - Los cambios son propios del usuario. Diferentes usuarios pueden setear su configuracion para la pantalla.
  - No es necesario tener privilegios de administrador para cambiar la resolución de la pantalla.

Posibles soluciones

Primera solución:

Cada vez que se inicie sesión, el archivo `$HOME/.profile` es invocado. En él se guardan las configuraciones personalizadas de cada usuario, de modo que cada uno de ellos tiene permiso de escritura.

Acciones a realizar:

- Bajar el contenido de la variable `cmd->str` directamente al archivo `$HOME/.profile`.
- Desde la función `set_xrandr_info()` se detectarían la ubicación de la carpeta home mediante la invocación de la función `getenv()`.

Problema:

- Cada vez que se llame a la función `set_xrandr_info()` se agregará una llamadaa xrandr.

Mejora:

- Realizar una busqueda en el archivo `$HOME/.profile` por una llamada a xrandr y sobreescribirla.

Posibles soluciones

Segunda solución:

- Para evitar las búsquedas dentro del archivo `$HOME/.profile` se puede agregar una sola vez una línea que ejecute un script en `$HOME/.config` donde están las configuraciones de LXDE.
- Cada vez que se llama a la función `set_xrandr_info()` se modifica solamente el script y no el archivo `$HOME/.profile`.

Solución implementada:

- El archivo `$HOME/.profile` contiene una línea nueva:

  `./xrandr.sh`

  al final de su contenido

- El llamado al script xrandr.sh se hace en los casos en que el usuario no se haya logueado desde una terminal real, porque en esa situación, se produciría un error debido a que la variable `$DISPLAY` no está definida.

static void set_xrandr_info() modificada

```
{
    [...]

    g_spawn_command_line_sync( cmd->str, NULL, NULL, NULL, NULL );
      save_xrand_info(cmd->str);
    g_string_free( cmd, TRUE );
}
```

Se agrega la llamada a la función **save_xrand_info(cmd->str)**

En en campo **str** de la estructura **cmd** se almacena un string con el comando para el **xrandr**

save_xrand_info(char * informacion)

/*This function save the resolution in $HOME/.config/lxrandr.sh */

```
FILE *fprof, *fscript;
char * home=NULL;
char archivo[512];
char profile[512];

 home = getenv("HOME");
 strcpy(archivo, home);
 strcpy(profile, home);
 strcat(profile, "/.profile");
 strcat(archivo, "/.config/lxrandr.sh");
```

```
    if ((fprof = fopen(profile, "a")) == NULL) {
    printf("Error, no se puede abrir %s\n", profile);
}
else{
        if ((fscript = fopen(archivo, "w")) == NULL){

                fprintf(fprof, "if [ -n \"$DISPLAY \" ]; then\n");
                fprintf(fprof, "\t%s%s\n", ". ", archivo);
                fprintf(fprof, "fi\n");
                fprintf(fscript, "#!/bin/bash\n");
                fprintf(fscript,"%s\n", informacion);
                fclose(fscript);
            }
        else{printf("Error, no se puede crear  %s\n", archivo);}

            fclose(fprof);
    }

}   /* fuctio */
```

Modificación lxrandr

Casos de prueba

Se comprobó funcionamiento de las modificaciones sobre

- Debian Lenny con LXDE
- Debian squeeze con LXDE

Además se envió un parche a los desarrolladores de LXDE, invitando a que incluyan estas modificaciones en la próxima versión de lxrandr.

Preguntas

???

Gracias!!

Info de contacto

Andrea Gómez Del Mónaco
Joaquín Bogado
soportelihuen [at] linti.unlp.edu.ar