

lazarus 使用技巧集

2022 年 6 月 (QQ: 315795176 秋风)

说明:

这是本人使用 **lazarus for linux aarch64** 过程遇到的问题及解决办法, 本文大部份内容是原创并记录在博客园 (<https://www.cnblogs.com/qiufeng2014>)。

● 1、用 **lazarus** 创建 **linux** 的菜单、桌面快捷方式及文件关联

为 lazarus 生成的 linux 程序提供相关的快捷访问方式, 参考 fpcupdeluxe 源码, 编写了一个通用的 CreateDesktopShortCut, 只要调用 CreateDesktopShortCut 就可以生成相应的快捷方式及文件关联。

开始菜单程序名称存放路径:

/home/用户名/.local/share/applications

文件关联放以下 2 个路径:

/home/用户名/.local/share/mime/application

/home/用户名/.local/share/mime/packages

桌面快捷方式存放路径:

/home/用户名/桌面

```
unit Unit1;
{$mode objfpc}{$H+}
interface
uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls, Buttons,
  IniFiles, BaseUnix, FileUtil, process
;
type
  { TForm1 }

TForm1 = class(TForm)
  Button1: TButton;
  Edit1: TEdit;
  Edit2: TEdit;
  Edit3: TEdit;
  Edit5: TEdit;
  Edit7: TEdit;
  Edit6: TEdit;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  OpenFileDialog1: TOpenDialog;
  SpeedButton1: TSpeedButton;
  SpeedButton2: TSpeedButton;
  procedure Button1Click(Sender: TObject);
  procedure SpeedButton1Click(Sender: TObject);
  procedure SpeedButton2Click(Sender: TObject);
private
public
end;
var
  Form1: TForm1;
implementation
{$R *.lfm}
{ TForm1 }
function ForceDirectoriesSafe(Const Dir: RawByteString): Boolean;
var
  aDir:RawByteString;
begin
```

```

result:=true;
if (Length(Dir)=0) then exit;
aDir:=ExcludeTrailingPathDelimiter(Dir);
if (Length(aDir)=0) then exit;
if (NOT DirectoryExists(aDir)) then
    result:=ForceDirectories(aDir);
end;
procedure CreateDesktopShortCut(Target,Execs,icons,filetypes,FileExt, ShortcutName: string);
var
    OperationSucceeded: boolean;
    ResultCode: boolean;
    XdgDesktopContent: TStringList;
    XdgMimeContent: TStringList;
    Output,XdgDesktopFile,XdgMimeFile: string;
    aDirectory:string;
    i,j:integer;
    AddContext:boolean;
    ft:TStrings;
begin
    ft:=TStringList.Create;
    ft.Delimiter:=';';
    ft.DelimitedText:=FileExt;
    j:=ft.Count;
    if (j>0) and (filetypes<>'') then AddContext:=true
    ELSE addcontext:=false;

    // Fail by default:
    OperationSucceeded:=false;

    XdgDesktopFile:=IncludeTrailingPathDelimiter(GetTempDir(false))+shortcutname+'.desktop';
    XdgDesktopContent:=TStringList.Create;
    try
        XdgDesktopContent.Add(['Desktop Entry']);
        XdgDesktopContent.Add('Version=1.0');
        XdgDesktopContent.Add('Encoding=UTF-8');
        XdgDesktopContent.Add('Type=Application');
        XdgDesktopContent.Add('Icon='+icons);      XdgDesktopContent.Add('Exec='+execs+' %u');
        XdgDesktopContent.Add('Name='+ShortcutName);      XdgDesktopContent.Add('Category=Application;');
        XdgDesktopContent.Add('Categories=Application;Programming;');
        if AddContext then
            begin
                XdgDesktopContent.Add('MimeType=application/x-'+filetypes+'');
            end;
        try
            XdgDesktopContent.SaveToFile(XdgDesktopFile);
            FpChmod(XdgDesktopFile, &711); //rwx--x--x

            OperationSucceeded:=RunCommand('xdg-desktop-icon' ,['install','--novendor',XdgDesktopFile],Output,[poUsePipes,
            poStderrToOutPut]{$IF DEFINED(FPC_FULLVERSION) AND (FPC_FULLVERSION >= 30200)},swoHide{$ENDIF});

            OperationSucceeded:=RunCommand('xdg-desktop-menu' ,['install','--novendor',XdgDesktopFile],Output,[poUsePipes,
            poStderrToOutPut]{$IF DEFINED(FPC_FULLVERSION) AND (FPC_FULLVERSION >= 30200)},swoHide{$ENDIF});
        except
            OperationSucceeded:=false;
        end;

        if (true) then
            begin
                aDirectory:=ConcatPaths(['usr','share','applications']);
                if ( (FpGetuid=0) AND DirectoryExists(aDirectory) ) then
                    begin
                        FileUtil.CopyFile(XdgDesktopFile,aDirectory+DirectorySeparator+ExtractFileName(XdgDesktopFile),[]);
                    end
                else
                    begin
                        // Create shortcut directly on User-Desktop
                        aDirectory:=ConcatPaths([GetUserDir,'Desktop']);
                        if DirectoryExists(aDirectory) then
                            FileUtil.CopyFile(XdgDesktopFile,aDirectory+DirectorySeparator+ExtractFileName(XdgDesktopFile),[]);
                        // Create user menu item
                    end
                end
            end
        end
    end
end

```

```

if (NOT OperationSucceeded) then
begin
    aDirectory:=ConcatPaths([GetUserDir,'.local','share','applications']);
    if DirectoryExists(aDirectory) then
        FileUtil.CopyFile(XdgDesktopFile,aDirectory+DirectorySeparator+ExtractFileName(XdgDesktopFile),[]);
    end;
end;
end;
// Temp file is no longer needed....
try
    SysUtils.DeleteFile(XdgDesktopFile);
finally
    // Swallow, let filesystem maintenance clear it up
end;
finally
    XdgDesktopContent.Free;
    OperationSucceeded:=true;
end;

if (OperationSucceeded) then
begin
    aDirectory:=ConcatPaths([GetUserDir,'.local','share','applications']);
    OperationSucceeded:=RunCommand('update-desktop-database',[aDirectory],Output,[poUsePipes,
poStderrToOutPut]{$IF DEFINED(FPC_FULLVERSION) AND (FPC_FULLVERSION >= 30200)},swoHide{$ENDIF});
end;
if AddContext then
begin
    {$ifdef LCL}
    Application.ProcessMessages;
    {$endif}

    aDirectory:=ConcatPaths([GetUserDir,'.local','share','mime']);
    ForceDirectoriesSafe(aDirectory);

    //Create mime file associations
    XdgMimeFile:=IncludeTrailingPathDelimiter(GetTempDir(false))+shortcutname+'.xml';
    XdgMimeContent:=TStringList.Create;
    try
        XdgMimeContent.Add('<?xml version="1.0" encoding="UTF-8"?>');
        XdgMimeContent.Add('<mime-info xmlns="http://www.freedesktop.org/standards/shared-mime-info">');
        XdgMimeContent.Add('    <mime-type type="application/x-' + filetype + '>');
        XdgMimeContent.Add('        <comment>TEXT file</comment>');
        XdgMimeContent.Add('        <icon name="application-x-' + filetype + '>');
        XdgMimeContent.Add('        <glob-deleteall/>');
        for i:=0 to j-1 do
        begin
            XdgMimeContent.Add('            <glob pattern="'+ft[i]+'"/>');
        end;
        XdgMimeContent.Add('    </mime-type>');
        XdgMimeContent.Add('</mime-info>');
        aDirectory:=ConcatPaths([GetUserDir,'.local','share','mime','packages']);
        ForceDirectoriesSafe(aDirectory);
        XdgMimeContent.SaveToFile(XdgMimeFile);
        OperationSucceeded:=RunCommand('xdg-mime',['install','--novendor',XdgMimeFile],Output,[poUsePipes,
poStderrToOutPut]{$IF DEFINED(FPC_FULLVERSION) AND (FPC_FULLVERSION >= 30200)},swoHide{$ENDIF});
        SysUtils.DeleteFile(XdgMimeFile);
    finally
        XdgMimeContent.Free;
    end;

    //Process icon
    aDirectory:=ConcatPaths([GetUserDir,'.local','share','icons']);
    ForceDirectoriesSafe(aDirectory);
    OperationSucceeded:=RunCommand('xdg-icon-resource',['install','--novendor',
'--context','mimetypes','--size','64',icons,'application-x-' + filetype + ''],Output,
[poUsePipes, poStderrToOutPut]{$IF DEFINED(FPC_FULLVERSION) AND (FPC_FULLVERSION >=
30200)},swoHide{$ENDIF});

    //Update mime database
    aDirectory:=ConcatPaths([GetUserDir,'.local','share','mime']);

```

```

        OperationSucceeded:=RunCommand('update-mime-database' ,[aDirectory],Output,
        [poUsePipes, poStderrToOutPut]{$IF DEFINED(FPC_FULLVERSION) AND (FPC_FULLVERSION >=
30200)},swoHide{$ENDIF});
    end;
    ft.free;
end;

procedure TForm1.Button1Click(Sender: TObject);
var aDirectory,vFileName,Output,deskname:String;
begin
    deskname:=edit1.Text;  aDirectory := ConcatPaths([GetUserDir, '桌面']);
    if not DirectoryExists(aDirectory) then
        aDirectory := ConcatPaths([GetUserDir, 'Desktop']);
    CreateDesktopShortCut(aDirectory,edit2.text,edit5.text,Edit6.text,Edit7.text,deskname);
end;
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
    if opendialog1.Execute then
        Begin
            edit2.Text:= opendialog1.FileName;
            edit1.Text:=ExtractFileName(edit2.Text);
        end;
end;
procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
    if opendialog1.Execute then
        Begin
            edit5.Text:= opendialog1.FileName;
        end;
end;
end.

```


桌面图标生成器

软件文件名

桌面名称

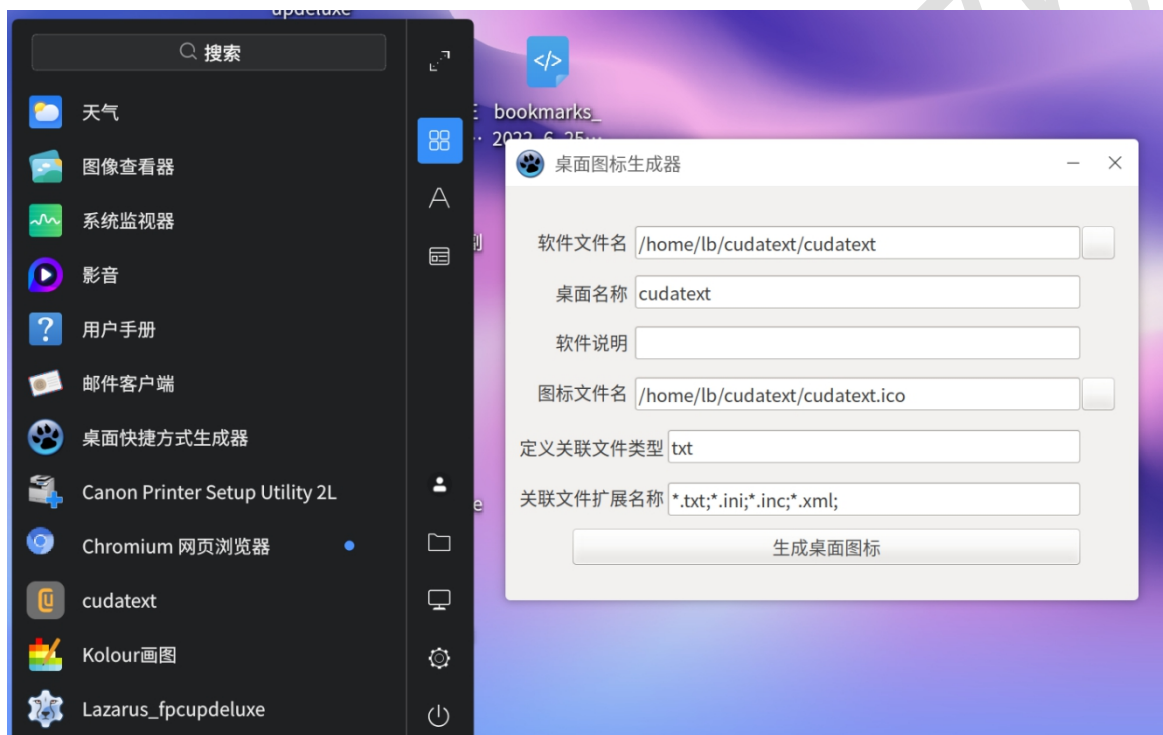
软件说明

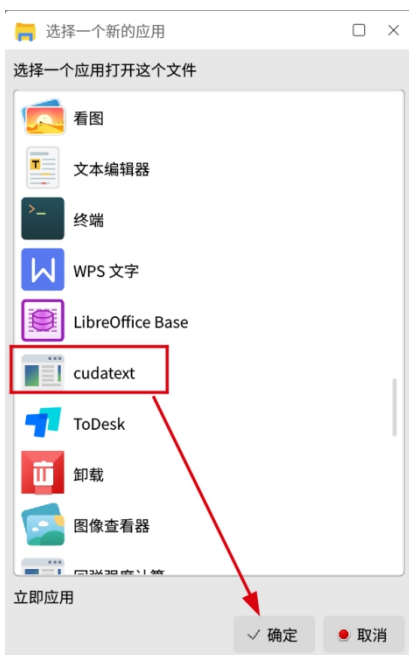
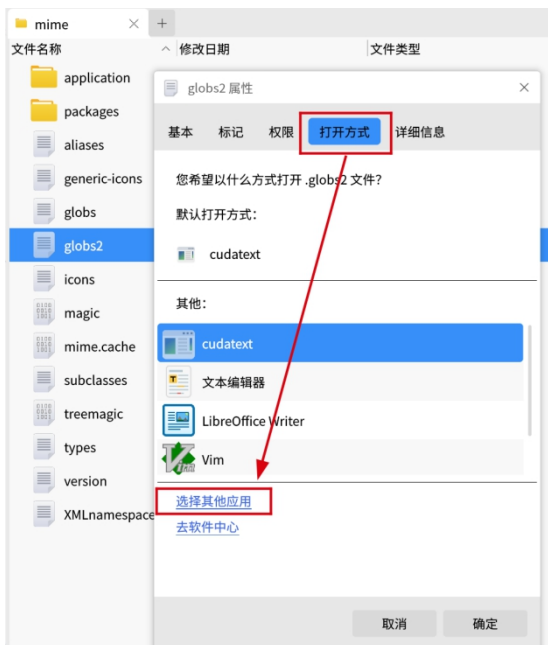
图标文件名

定义关联文件类型

关联文件扩展名称

生成桌面图标





● 2、fpc 源码 **clocale.pp** 在中文 linux 存在的 2 个问题
clocale.pp 在中文 linux 存在的 2 个问题

注：1、红色行是修改后的代码。

2、将以下代码另存为 **clocale.pp**

```
{
  This file is part of the Free Pascal run time library.
  Copyright (c) 2008 by the Free Pascal development team.

  Init rtl formatting variables based on libc locales

  See the file COPYING.FPC, included in this distribution,
  for details about the copyright.

  This program is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
  *****}
{ Initial implementation by petr kristan }
unit clocale;
{$ifdef android}
  {$error This unit is not intended for Android. Something wrong with the make file. }{$endif android}
{$mode objfpc}
interface
{$ifdef localedebug} // for easier debugging, allows to print untransformed values in test
Type TOrgFormatSettings = record
    ShortDateFormat,
    LongDateFormat ,
    ShortTimeFormat,
    LongTimeFormat ,
    CurrencyString1,
    CurrencyString2: string;
end;
var OrgFormatSettings : TOrgFormatSettings;
{$endif}
implementation
{$linklib c}

Uses
  SysUtils, unixtype, initc;

Const{$if defined(BSD) or defined(SUNOS) or defined(aix)}
  // Darwin, FreeBSD, Solaris, AIX. Note the lead underscores are added.
  {$i clocale.inc}{$else} // checked for Linux only, but might be general glibc.
    _LC_CTYPE      = 0;
    _LC_NUMERIC    = 1;
    _LC_TIME       = 2;
    _LC_COLLATE    = 3;
    _LC_MONETARY   = 4;
    _LC_MESSAGES   = 5;
    _LC_ALL        = 6;
    ABDAY_1 = (_LC_TIME shl 16);
    DAY_1 = (ABDAY_1)+7;
    ABMON_1 = (ABDAY_1)+14;
    MON_1 = (ABDAY_1)+26;
    AM_STR = (ABDAY_1)+38;
    PM_STR = (ABDAY_1)+39;
    D_T_FMT = (ABDAY_1)+40;
    D_FMT = (ABDAY_1)+41;
    T_FMT = (ABDAY_1)+42;
    T_FMT_AMPM = (ABDAY_1)+43;

    _DECIMAL_POINT = (_LC_NUMERIC shl 16);
    RADIXCHAR = _DECIMAL_POINT;
    _THOUSANDS_SEP = (_DECIMAL_POINT)+1;

    _INT_CURR_SYMBOL = (_LC_MONETARY shl 16);
    _CURRENCY_SYMBOL = (_INT_CURR_SYMBOL)+1;
    _MON_DECIMAL_POINT = (_INT_CURR_SYMBOL)+2;
    _MON_THOUSANDS_SEP = (_INT_CURR_SYMBOL)+3;
    _MON_GROUPING = (_INT_CURR_SYMBOL)+4;
    _POSITIVE_SIGN = (_INT_CURR_SYMBOL)+5;
```

```

_NEGATIVE_SIGN = (_INT_CURR_SYMBOL)+6;
_INT_FRAC_DIGITS = (_INT_CURR_SYMBOL)+7;
_FRAC_DIGITS = (_INT_CURR_SYMBOL)+8;
_P_CS_PRECEDES = (_INT_CURR_SYMBOL)+9;
_P_SEP_BY_SPACE = (_INT_CURR_SYMBOL)+10;
_N_CS_PRECEDES = (_INT_CURR_SYMBOL)+11;
_N_SEP_BY_SPACE = (_INT_CURR_SYMBOL)+12;
_P_SIGN_POSN = (_INT_CURR_SYMBOL)+13;
_N_SIGN_POSN = (_INT_CURR_SYMBOL)+14;
_NL_MONETARY_CRNCYSTR = (_INT_CURR_SYMBOL)+15;
#endif
#ifdef netbsd
{ NetBSD has a new setlocale function defined in /usr/include/locale.h
that should be used }function setlocale(category: cint; locale: pchar): pchar; cdecl; external clib name
'__setlocale_mb_len_max_32';{$else}function setlocale(category: cint; locale: pchar): pchar; cdecl; external clib name
'setlocale';{$endif}
function nl_langinfo(_item: cint):Pchar;cdecl;external clib name 'nl_langinfo';
procedure GetFormatSettings(out fmts: TFormatSettings);

function GetLocaleStr(item: cint): string;
begin
  GetLocaleStr := AnsiString(nl_langinfo(item));
end;

function GetLocaleChar(item: cint): char;
begin
  GetLocaleChar := nl_langinfo(item)^;
end;

function SkipModifiers(const s: string; var i: integer): string;
var
  l: Integer;
begin
  Result := "";
  l := Length(s);
  //possible flag, with specifier or modifier - glibc extension
  while (i<=l) and (s[i] in ['0'..'9', '_', '-', '^', '#', 'E', 'O']) do
  begin
    Result := Result + s[i];
    inc(i);
  end;
end;

function IsModifier(const Mods: string; m: char): boolean;
var
  i: integer;
begin
  Result := False;
  for i := 1 to Length(Mods) do begin
    if Mods[i] = m then begin
      Result := True;
      Break;
    end;
  end;
end;

function FindSeparator(const s: string; Def: char): char;
var
  i: integer;
begin
  FindSeparator := Def;
  i := Pos('%', s);
  if i=0 then
    Exit;
  inc(i);
  SkipModifiers(s, i);
  inc(i);
  if i<=Length(s) then
    FindSeparator:=s[i];
  if ord(FindSeparator)>127 then FindSeparator:=Def;
end;

```

```

end;

function TransformFormatStr(const s: string): string;
var
  i, l: integer;      ampminstring : boolean;    clock12:boolean;
  LastMod: string;
begin
  clock12:=false; // should ampm get appended?    ampminstring:=false;    TransformFormatStr := '';
  i := 1;
  l := Length(s);
  while i<=l do begin
    if s[i]='%' then begin
      inc(i);
      LastMod := SkipModifiers(s, i);
      if i>l then
        Exit;
      case s[i] of
        'a': TransformFormatStr := TransformFormatStr + 'ddd';
        'A': TransformFormatStr := TransformFormatStr + 'dddd';
        'b': TransformFormatStr := TransformFormatStr + 'mmm';
        'B': TransformFormatStr := TransformFormatStr + 'mmm';
        'c': TransformFormatStr := TransformFormatStr + 'c';
        // 'C':
        'd': if IsModifier(LastMod, '-') then
              TransformFormatStr := TransformFormatStr + 'd'
            else
              TransformFormatStr := TransformFormatStr + 'dd';
        'D': TransformFormatStr := TransformFormatStr + 'mm'/'dd'/'yy';
        'e': TransformFormatStr := TransformFormatStr + 'd';
        'F': TransformFormatStr := TransformFormatStr + 'yyyy-mm-dd';
        'g': TransformFormatStr := TransformFormatStr + 'yy';
        'G': TransformFormatStr := TransformFormatStr + 'yyyy';
        'h': TransformFormatStr := TransformFormatStr + 'mmm';
        'H': TransformFormatStr := TransformFormatStr + 'hh';
        'I': begin
              TransformFormatStr := TransformFormatStr + 'hh';
              clock12:=true;
            end;
        // 'j':
        'k': TransformFormatStr := TransformFormatStr + 'h';
        'l': begin
              TransformFormatStr := TransformFormatStr + 'h';
              clock12:=true;
            end;
        'm': if IsModifier(LastMod, '-') then
              TransformFormatStr := TransformFormatStr + 'm'
            else
              TransformFormatStr := TransformFormatStr + 'mm';
        'M': TransformFormatStr := TransformFormatStr + 'nn';
        'n': TransformFormatStr := TransformFormatStr + sLineBreak;
        'p','P':
          begin
            TransformFormatStr := TransformFormatStr + 'ampm';
            ampminstring:=true;
            clock12:=false;
          end;
        'r': begin
              TransformFormatStr := TransformFormatStr + 'hh:nn:ss';
              clock12:=true;
            end;
        'R': TransformFormatStr := TransformFormatStr + 'hh:nn';
        // 's':
        'S': TransformFormatStr := TransformFormatStr + 'ss';
        't': TransformFormatStr := TransformFormatStr + #9;
        'T': TransformFormatStr := TransformFormatStr + 'hh:nn:ss';
        // 'u':
        // 'U':
        // 'V':
        // 'w':
        // 'W':
      end;
      i := i + 1;
    end;
  end;
end;

```

```

        'x': TransformFormatStr := TransformFormatStr + 'dddd';
        'X': TransformFormatStr := TransformFormatStr + 't';
        'y': TransformFormatStr := TransformFormatStr + 'yy';
        'Y': TransformFormatStr := TransformFormatStr + 'yyyy';
        //'z':
        //'Z':
        '%': TransformFormatStr := TransformFormatStr + '%';
    end;
end else
    TransformFormatStr := TransformFormatStr + s[i];
    inc(i);
end;
if ampminstring then
    clock12:=false;
i:=length(TransformFormatStr);
if clock12 and (i>0) then
    begin
        if transformformatstr[i]<>' ' then
            TransformFormatStr := TransformFormatStr + ' ';
            TransformFormatStr := TransformFormatStr + 'ampm';
        end;
    end;
end;
const
    //          sign  prec  sep
    NegFormatsTable: array [0..4, 0..1, 0..1] of byte = (
        ( (4, 15), (0, 14) ), //Parentheses surround the quantity and currency_symbol
        ( (5, 8), (1, 9) ), //The sign string precedes the quantity and currency_symbol
        ( (7, 10), (3, 11) ), //The sign string follows the quantity and currency_symbol
        ( (6, 13), (1, 9) ), //The sign string immediately precedes the currency_symbol
        ( (7, 10), (2, 12) ) //The sign string immediately follows the currency_symbol ); var
i: integer;
prec, sep, signp: byte;
{$if defined(BSD) or defined(SUNOS) or defined(aix)}
    locale : plconv;
{$ENDIF}begin
setlocale(__LC_ALL,"");
for i := 1 to 12 do
    begin
        fmts.ShortMonthNames[i]:=GetLocaleStr(ABMON_1+i-1);
        fmts.LongMonthNames[i]:=GetLocaleStr(MON_1+i-1);
    end;
for i := 1 to 7 do
    begin
        fmts.ShortDayNames[i]:=GetLocaleStr(ABDAY_1+i-1);
        fmts.LongDayNames[i]:=GetLocaleStr(DAY_1+i-1);
    end;
//Date stuff
fmts.ShortDateFormat := GetLocaleStr(D_FMT);
{$ifdef localedebug}
    OrgFormatSettings.ShortDateFormat:=fmts.shortdateformat;{$endif}

fmts.DateSeparator := FindSeparator(fmts.ShortDateFormat, fmts.DateSeparator);
fmts.ShortDateFormat := TransformFormatStr(fmts.ShortDateFormat);
fmts.LongDateFormat := GetLocaleStr(D_FMT);{$ifdef localedebug}
    OrgFormatSettings.LongDateFormat:=fmts.longdateformat;{$endif}
fmts.LongDateFormat := TransformFormatStr(fmts.LongDateFormat);
//Time stuff
fmts.TimeAMString := GetLocaleStr(AM_STR);
fmts.TimePMString := GetLocaleStr(PM_STR);
fmts.ShortTimeFormat := GetLocaleStr(T_FMT);{$ifdef localedebug}
    OrgFormatSettings.ShortTimeFormat:=fmts.shorttimeformat;{$endif}
fmts.TimeSeparator := FindSeparator(fmts.ShortTimeFormat, fmts.TimeSeparator);
fmts.ShortTimeFormat := TransformFormatStr(fmts.ShortTimeFormat);
fmts.LongTimeFormat := GetLocaleStr(T_FMT_AMP);{$ifdef localedebug}
    OrgFormatSettings.LongTimeFormat:=fmts.longtimeformat;{$endif}

if (fmts.LongTimeFormat='') then
    fmts.LongTimeFormat:=fmts.ShortTimeFormat
else
    fmts.LongTimeFormat := TransformFormatStr(fmts.LongTimeFormat);

```



```

{$if defined(BSD) or defined(SUNOS) or defined(aix)}
    plocale:=localeconv;
    // for these fields there is a separate BSD derived POSIX function.
    if not assigned(plocale) then exit; // for now.
    fmts.CurrencyString:=plocale^.currency_symbol; // int_CURR_SYMBOL (in latin chars)
    if fmts.CurrencyString="" then
        fmts.CurrencyString:=plocale^.int_curr_symbol;
    fmts.CurrencyDecimals:=ord(plocale^.FRAC_DIGITS); {$ifdef localedebug}
OrgFormatSettings.CurrencyString1:=plocale^.currency_symbol;
OrgFormatSettings.CurrencyString2:=plocale^.int_curr_symbol; {$endif}
    prec:=ord(plocale^.P_CS_PRECEDES);
    sep:=ord(plocale^.P_SEP_BY_SPACE);
    if (prec<=1) and (sep<=1) then
        fmts.CurrencyFormat := byte(not boolean(prec)) + sep shl 1;
    prec := ord(plocale^.N_CS_PRECEDES);
    sep := ord(plocale^.N_SEP_BY_SPACE);
    signp := ord(plocale^.N_SIGN_POSN);
    if (signp in [0..4]) and (prec in [0, 1]) and (sep in [0, 1]) then
        fmts.NegCurrFormat := NegFormatsTable[signp, prec, sep];
//Number stuff
    fmts.ThousandSeparator:=plocale^.THOUSANDS_SEP[0];
{$else}
    //Currency stuff
    fmts.CurrencyString := GetLocaleStr(_NL_MONETARY_CRNCYSTR); {$ifdef localedebug}
OrgFormatSettings.CurrencyString1:=fmts.currencystring;
OrgFormatSettings.CurrencyString2:=""; {$endif}
    fmts.CurrencyString := Copy(fmts.CurrencyString, 2, Length(fmts.CurrencyString));
    fmts.CurrencyDecimals := StrToIntDef(GetLocaleStr(_FRAC_DIGITS), fmts.CurrencyDecimals);
    prec := byte(GetLocaleChar(_P_CS_PRECEDES));
    sep := byte(GetLocaleChar(_P_SEP_BY_SPACE));
    if (prec<=1) and (sep<=1) then
        fmts.CurrencyFormat := byte(not boolean(prec)) + sep shl 1;
    prec := byte(GetLocaleChar(_N_CS_PRECEDES));
    sep := byte(GetLocaleChar(_N_SEP_BY_SPACE));
    signp := byte(GetLocaleChar(_N_SIGN_POSN));
    if (signp in [0..4]) and (prec in [0, 1]) and (sep in [0, 1]) then
        fmts.NegCurrFormat := NegFormatsTable[signp, prec, sep];
//Number stuff
    fmts.ThousandSeparator:=GetLocaleChar(_THOUSANDS_SEP);
    Sep := ord(GetLocaleChar(_MON_THOUSANDS_SEP));
    if fmts.ThousandSeparator=#0 then
        fmts.ThousandSeparator := char(Sep);
    {$endif}
    fmts.DecimalSeparator:=GetLocaleChar(RADIXCHAR);end;
initialization
    GetFormatSettings(DefaultFormatSettings);
end.

```

● 3、lazarus 使用建议

在 lazarus 过程中, 发现很多问题与 DefaultFormatSettings 有关(特别是在 linux 中使用)感觉 lazarus 没将 DefaultFormatSettings 赋予初始值, 如 mask 字符中包含/: 会出现乱码、unidac 日期字段和浮点等问题。建议在 project.lpr 增加 DefaultFormatSettings 初始值避免程序出现各种异常, 可以参考以下红色的代码。

```

program acs;
{$mode objfpc}{$H+}
uses
    {$IFDEF UNIX}
    cthreads,
    {$ENDIF}
    Interfaces, SysUtils, //要加这个单元 Forms,
    main ;
{$R *.res}
begin
    DefaultFormatSettings.ShortTimeFormat:='yyyy-mm-dd';
    DefaultFormatSettings.ShortTimeFormat:='hh:NN:ss';
    DefaultFormatSettings.LongDateFormat:='yyyy-mm-dd';

```



```

DefaultFormatSettings.LongTimeFormat:='hh:NN:ss';
DefaultFormatSettings.DateSeparator:='-';
DefaultFormatSettings.TimeSeparator:=':.'; RequireDerivedFormResource := True;
Application.Initialize;
Application.CreateForm(TMainForm, MainForm);
Application.Run;
end.

```

● 4、lazarus 读取版本信息(适用于 windows/linux/macos)

从官网论坛看到的

(https://wiki.freepascal.org/Show_Application_Title,_Version,_and_Company#FPC_3.0.2B)。

下面的方法适用 FPC3.0 以上版本：

```

program printfileinfo;
{
  Displays file version info for
- Windows PE executables
- Linux ELF executables (compiled by Lazarus)
- macOS MACH-O executables (compiled by Lazarus)
  Runs on Windows, Linux, macOS}
{$mode objfpc}{$H+}{$ifdef mswindows}{$apptype console}{$endif}uses
  {$IFDEF UNIX}{$IFDEF UseCThreads}
  cthreads,
  {$ENDIF}{$ENDIF}
  Classes,sysutils
  // FPC 3.0 fileinfo reads exe resources as long as you register the appropriate units , fileinfo
  , winpeimagereader {need this for reading exe info}
  , elfreader {needed for reading ELF executables}
  , machoreader {needed for reading MACH-O executables}
;
var
  FileVerInfo: TFileVersionInfo;
{$R *.res}
begin
  FileVerInfo:=TFileVersionInfo.Create(nil);
  try
    FileVerInfo.ReadFileInfo;
    writeln('Company: ',FileVerInfo.VersionStrings.Values['CompanyName']);
    writeln('File description: ',FileVerInfo.VersionStrings.Values['FileDescription']);
    writeln('File version: ',FileVerInfo.VersionStrings.Values['FileVersion']);
    writeln('Internal name: ',FileVerInfo.VersionStrings.Values['InternalName']);
    writeln('Legal copyright: ',FileVerInfo.VersionStrings.Values['LegalCopyright']);
    writeln('Original filename: ',FileVerInfo.VersionStrings.Values['OriginalFilename']);
    writeln('Product name: ',FileVerInfo.VersionStrings.Values['ProductName']);
    writeln('Product version: ',FileVerInfo.VersionStrings.Values['ProductVersion']);
  finally
    FileVerInfo.Free;
  end;
end.

```

FPC3.0 之前版本使用下面的方法读取版本信息

```

uses
  resource, versiontypes, versionresource;

FUNCTION resourceVersionInfo: STRING;

(* Unlike most of AboutText (below), this takes significant activity at run- *)
(* time to extract version/release/build numbers from resource information *)
(* appended to the binary. *)

VAR
  Stream: TResourceStream;
  vr: TVersionResource;
  fi: TVersionFixedInfo;

BEGIN
  RESULT:= '';
  TRY

```

```
(* This raises an exception if version info has not been incorporated into the *)
(* binary (Lazarus Project -> Project Options -> Version Info -> Version *)
(* numbering). *)
```

```
Stream:= TResourceStream.CreateFromID(HINSTANCE, 1, PChar(RT_VERSION));
TRY
  vr:= TVersionResource.Create;
  TRY
    vr.SetCustomRawDataStream(Stream);
    fi:= vr.FixedInfo;
    RESULT := 'Version ' + IntToStr(fi.FileVersion[0]) + '.' + IntToStr(fi.FileVersion[1]) +
      ' release ' + IntToStr(fi.FileVersion[2]) + ' build ' + IntToStr(fi.FileVersion[3]) + eol;
    vr.SetCustomRawDataStream(nil)
  FINALLY
    vr.Free
  END
  Stream.Free
END
EXCEPT
  END
END { resourceVersionInfo };
```

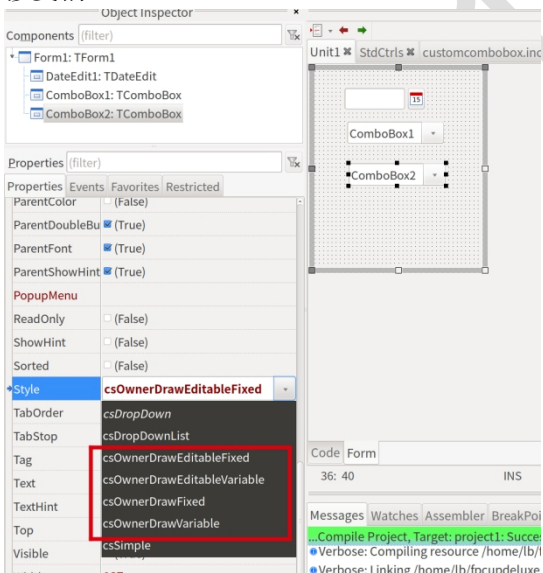
● 5、修复 lazarus linux(ubuntu/银河麒麟) Object Inspector、使用 combobox、colorbox 等控件下拉列表文字不显示的问题

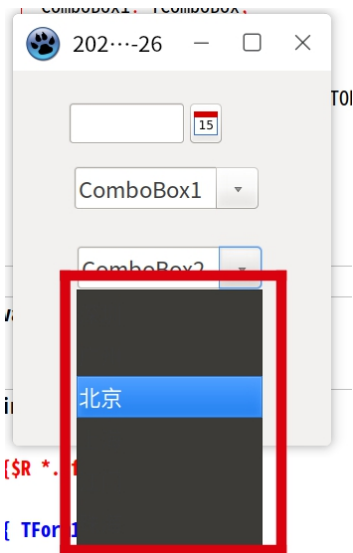
修复 lazarus linux(ubuntu/银河麒麟)Object Inspector、使用 combobox、colorbox 等控件 style 为 csOwnerDraw*时下拉列表文字不显示的问题（在树莓派正常），这类问题大概率是 ubuntu/银河麒麟引起的，如果使用中也遇到同样的问题可参照以下方法处理。

打开 lazarus/lcl/include/customcombobox.inc, (第 109 行)定位 TCustomComboBox.DrawItem(Index: Integer; ARect: TRect; State: TOwnerDrawState);

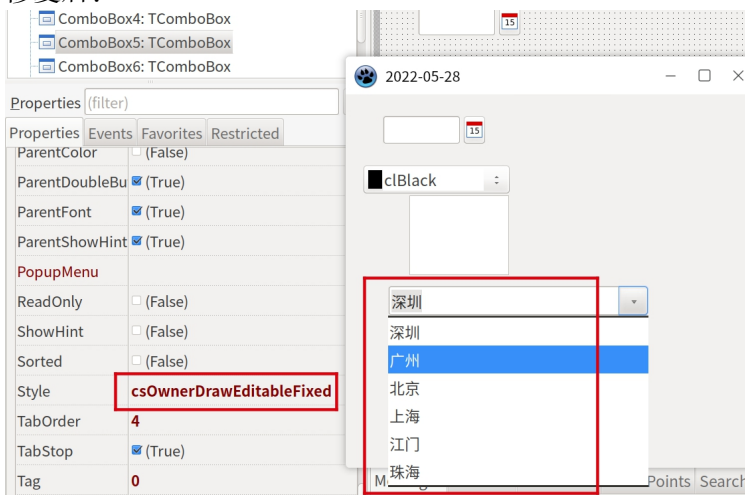
添加红字 1 行，重绘背景，重新编译应用程序就可以解决 Object Inspector、combobox 和 colorbox 列表的不显示 item 的问题，这个方法不一定是最佳的，但能解决我遇到的问题。

修复前：





修复后：



```

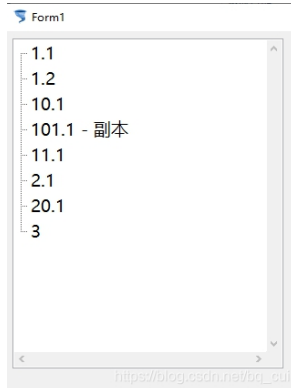
procedure TCustomComboBox.DrawItem(Index: Integer; ARect: TRect;
  State: TOwnerDrawState);
begin
  //TControlCanvas(FCanvas).UpdateTextFlags;
  //2022.05.28 LBZ
  {$ifdef linux}
  FCanvas.FillRect(ARect);
  {$endif}
  //2022.05.28 LBZ
  if Assigned(FOnDrawItem) then
    FOnDrawItem(Self, Index, ARect, State)
  else
    begin
      if not (odBackgroundPainted in State) then
        FCanvas.FillRect(ARect)
      else
        InternalDrawItem(Self, FCanvas, ARect, Items[Index]);
    end;
end;

```

● **6、【转】lazarus：对 treeview 控件内容进行自然排序**
有以下文件夹：



用 lazarus 中的 treeview 控件显示，默认是这样的：



现在我们需要按从小到大顺序排列。

先建一个 **natural** 模块：

```
unit natural;
{$MODE OBJFPC}{$H+}
// Natural Order String Comparison by Martin Pool
(* *- mode: c; c-file-style: "k&r" *-

strnatcmp.c -- Perform 'natural order' comparisons of strings in C.
Copyright (C) 2000, 2004 by Martin Pool <mbp sourcefrog net>

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would be
   appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
   misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

interface(* CUSTOMIZATION SECTION
*
* You can change this typedef, but must then also change the inline
* functions in strnatcmp.c *)
type
  nat_char = char;
  pnat_char = ^nat_char;
  function strnatcmp(const a: pnat_char; const b: pnat_char): integer;
  function strnatcasecmp(const a: pnat_char; const b: pnat_char): integer;
  function IsDigits and IsSpace(* function IsDigit(ch: Char): Boolean;
begin
  Result := ch in ['0'..'9'];
end;
function IsSpace(ch: Char): Boolean;
begin
  Result := ch in [' ', #9, #10, #11, #12, #13];
end;
(* partial change history:
*
```

* 2004-10-10 mbp: Lift out character type dependencies into macros.
 *
 * Eric Sosman pointed out that ctype functions take a parameter whose
 * value must be that of an unsigned int, even on platforms that have
 * negative chars in their default char type.
 *)

(* These are defined as macros to make it easier to adapt this code to
 * different characters types or comparison functions. *)

function nat_isdigit(a: nat_char): boolean; inline;

begin

result := IsDigit(char(a));

end;

function nat_isspace(a: nat_char): boolean; inline;

begin

result := IsSpace(char(a));

end;

function nat_toupper(a: nat_char): nat_char; inline;

begin

result := UpCase(char(a));

end;

function compare_right(a: pnat_char; b: pnat_char): integer;

var

bias : integer = 0;

begin

(* The longest run of digits wins. That aside, the greatest
 value wins, but we can't know that it will until we've scanned
 both numbers to know that they have the same magnitude, so we
 remember it in BIAS. *)

while true do

begin

if (not nat_isdigit(a^) and not nat_isdigit(b^))

then exit(bias)

else if (not nat_isdigit(a^))

then exit(-1)

else if (not nat_isdigit(b^))

then exit(1)

else if (a^ < b^) then

begin

if bias <> 0 then bias := -1;

end

else if (a^ > b^) then

begin

if bias <> 0 then bias := 1;

end

else if (a^ = #0) and(b^ = #0)

then exit(bias);

inc(a);

inc(b);

end;

result := 0;

end;

function compare_left(a: pnat_char; b: pnat_char): integer;

begin

(* Compare two left-aligned numbers: the first to have a
 different value wins. *)

while true do

begin

if (not nat_isdigit(a^) and not nat_isdigit(b^))

then exit(0)

else if (not nat_isdigit(a^))

then exit(-1)

else if (not nat_isdigit(b^))

then exit(1)

else if (a^ < b^)

then exit(-1)

else if (a^ > b^)

then exit(1);

inc(a);

inc(b);

```

end;
result := 0;
end;
function strnatcmp0(const a: pnat_char; const b: pnat_char; fold_case: integer): integer;
var
  ai, bi: integer;
  ca, cb: char;
  fractional: boolean;
begin
  assert( (a <> nil) and (b <> nil));
  ai := 0; bi := 0;
  while true do
    begin
      ca := a[ai];
      cb := b[bi];
      // skip over leading spaces or zeros
      while nat_isspace(ca) do
        begin
          inc(ai);
          ca := a[ai];
        end;
      while nat_isspace(cb) do
        begin
          inc(bi);
          cb := b[bi];
        end;
      // process run of digits
      if (nat_isdigit(ca) and nat_isdigit(cb)) then
        begin
          fractional := ((ca = '0') or (cb = '0'));
          if fractional then
            begin
              result := compare_left(a+ai, b+bi);
              if result <> 0 then exit;
            end
          else
            begin
              result := compare_right(a+ai, b+bi);
              if result <> 0 then exit;
            end;
          end;
        if (ca=#0) and (cb=#0) then
          begin
            (* The strings compare the same. Perhaps the caller
               will want to call strcmp to break the tie. *)
            exit(0);
          end;
        if fold_case <> 0 then
          begin
            ca := nat_toupper(ca);
            cb := nat_toupper(cb);
          end;
        if (ca < cb)
          then exit(-1)
        else if (ca > cb)
          then exit(1);
        inc(ai);
        inc(bi);
      end;
    end;
  end;

function strnatcmp(const a: pnat_char; const b: pnat_char): integer;
begin
  result := strnatcmp0(a, b, 0);
end;

(* Compare, recognizing numeric string and ignoring case. *)
function strnatcasecmp(const a: pnat_char; const b: pnat_char): integer;
begin
  result := strnatcmp0(a, b, 1);
end;

```

end;

end.

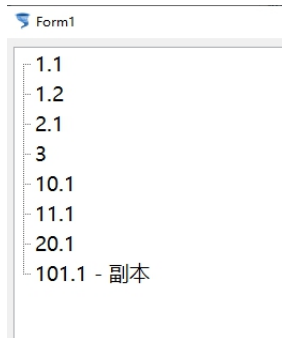
在主程序中，建立一个过程：

```
function TForm1.TreeviewAlphaSort(Node1, Node2: TTreeNode): Integer;
var
  a, b: pnat_char;
begin
  a := pnat_char(Node1.Text);
  b := pnat_char(Node2.Text);
  Result := strnatcmp(a, b)
end;
```

调用该过程：

TreeView1.CustomSort(@TreeviewAlphaSort);

运行结果：



完整代码：

```
unit Unit1;
{$mode objfpc}{$H+}
interface
uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, ComCtrls, StdCtrls, natural, LazFileUtils;
type
  { TForm1 }

TForm1 = class(TForm)
  Memo1: TMemo;
  TreeView1: TTreeView;
  procedure FormCreate(Sender: TObject);
  procedure FormDestroy(Sender: TObject);
private
  function TreeviewAlphaSort(Node1, Node2: TTreeNode): Integer;
public
end;
var
  Form1: TForm1;

function IsEmptyDir(sDir: String): Boolean;
function AttachMentsExists(FileName: String): Boolean;
procedure SetIcons(TreeView1: TTreeView; list: TStringList);
procedure EnumText(s: string; altem: TTreeNode);
procedure DirToTreeView(Tree: TTreeView; Directory: string; Root: TTreeNode;
  IncludeFiles: Boolean; FileExt: string);
function ExtractTreeViewFileName(RootPath: string; TreeView: TTreeView;
  FileExt: string): string;
function ExtractNodeFullPath(TreeView: TTreeView): string;
implementation
{$R *.frm}
var
  list: TStringList;
  RootPath: string; // = 'D:\C++Builder 学习大全中文版';
```

```

//FileName: string;
{ TForm1 }
function ExtractNodeFullPath(TreeView: TTreeView): string;
var
    Path: string;
    Parent: TTreeNode;
    // Node: TTreeNode;
begin
    Path := TreeView.Selected.text;
    Parent := TreeView.Selected.Parent;
    while Parent <> nil do
    begin
        Path := Parent.text + '\' + Path;
        Parent := Parent.Parent;
    end;
    Result := Path;end;
function ExtractTreeViewFileName(RootPath: string; TreeView: TTreeView;
    FileExt: string): string;var
    FileName: string;begin
    Result := "";
    if TreeView.Selected = nil then
        Exit;
    FileName := RootPath + ExtractNodeFullPath(TreeView) + FileExt; // 当前选中的文件名

    if not FileExists(FileName) then
        Exit;
    Result := FileName;end;
{
    将 1 个目录里面所有的文件添加到 TREEVIEW 中
    DirToTreeView(TreeView1,'D:\Data',nil,True,'.cpp');}
procedure DirToTreeView(Tree: TTreeView; Directory: string; Root: TTreeNode;
    IncludeFiles: Boolean; FileExt: string);
var
    SearchRec: TSearchRec;
    ItemTemp: TTreeNode;
begin
    with Tree.Items do
    begin
        BeginUpdate;
        if Directory[Length(Directory)] <> '\' then
            Directory := Directory + '\';
        if FindFirst(Directory + '*.*', faDirectory, SearchRec) = 0 then
        begin
            Application.ProcessMessages;
            repeat
                { 添加文件夹 }
                if (SearchRec.Attr and faDirectory = faDirectory) and
                    (SearchRec.Name[1] <> '.') then
                begin
                    if (RightStr(SearchRec.Name, 6) = '_files') or // 不添加 _file 这个文件夹
                        (RightStr(SearchRec.Name, 12) = '_Attachments') then
                        // 不添加 _AttachMents 这个文件夹
                            Continue;

                    if (SearchRec.Attr and faDirectory > 0) then
                        Root := AddChild(Root, SearchRec.Name);

                    ItemTemp := Root.Parent;

                    DirToTreeView(Tree, Directory + SearchRec.Name, Root,
                        IncludeFiles, FileExt);
                    Root := ItemTemp;
                end

                { 添加文件 }
            else if IncludeFiles then
                if SearchRec.Name[1] <> '.' then
                    if (RightStr(SearchRec.Name, 4) = FileExt) (* or { 只添加 .CPP 格式文件 }
                        (RightStr(SearchRec.Name, 4) <> '') *) then { 什么格式都添加 }

```



```

        AddChild(Root, SearchRec.Name);

        until FindNext(SearchRec) <> 0;
        FindClose(SearchRec);

    end;
    EndUpdate;
end;
end;

procedure EnumText(s: string; altem: TTreeNode);
var
    node: TTreeNode;
    str: string;
begin
    node := altem;
    while node <> nil do
    begin
        if s = "" then
            str := node.text
        else
            str := s + '\' + node.text;
        list.Add('----'+str);
        if node.HasChildren then
            EnumText(str, node.getFirstChild);

        node := node.getNextSibling;
    end;
end;

function IsEmptyDir(sDir: String): Boolean;
var
    sr: TSearchRec;
begin
    Result := true;
    if Copy(sDir, Length(sDir) - 1, 1) <> '\' then
        sDir := sDir + '\';
    if FindFirst(sDir + '*.*', faAnyFile, sr) = 0 then
        repeat
            if (sr.Name <> '.') and (sr.Name <> '..') then
            begin
                Result := False;
                break;
            end;
        until FindNext(sr) <> 0;
    FindClose(sr);end;
{
返回 附件文件夹
"D:\C++Builder 学习大全中文版\新建文本文档.htm"
D:\C++Builder 学习大全中文版\新建文本文档_Attachments}
function AttachmentsFolder(FileName: String): string;
begin
    Result := ExtractFilePath(FileName) + ChangeFileExt(ExtractFileName(FileName),
        "") + '_Attachments';
end;
function AttachMentsExists(FileName: String): Boolean;
var
    f: string;
begin
    f := ExtractFilePath(FileName) + ChangeFileExt(ExtractFileName(FileName), "")
        + '_Attachments';
    Result := DirectoryExists(f);
end;
procedure SetIcons(TreeView1: TTreeView; list: TStringList);
var
    i: Integer;
begin
    with TreeView1 do
    begin

```

```

for i := 0 to Items.Count - 1 do
begin
    if DirectoryExists(list.Strings[i]) then
    begin
        Items[i].ImageIndex := 0;
        Items[i].SelectedIndex := 0;
        Items[i].StateIndex := 0;
    end;

    {
    // 以下代码处理文件
    if FileExists(list.Strings[i]) then
    begin
        Items[i].ImageIndex := 1;
        Items[i].SelectedIndex := 1;
        Items[i].StateIndex := 1;
    end;

    // 以下代码处理带附件文件
    if (AttachMentsExists(list.Strings[i])) then
    if not IsEmptyDir( AttachmentsFolder(list.Strings[i]) ) then
    begin
        // Form1.Memo1.LINES.Add( AttachmentsFolder(list.Strings[i]));
        Items[i].ImageIndex := 2;
        Items[i].SelectedIndex := 2;
        Items[i].StateIndex := 2;
    end;
    }

end;
end;
end;

function TForm1.TreeviewAlphaSort(Node1, Node2: TTreeNode): Integer;
var
    a, b: pnat_char;
begin
    //PChar(Node1.Text), PChar(Node2.Text)
    a := pnat_char(Node1.Text);
    b := pnat_char(Node2.Text);

    //a := pnat_char(ExtractFileNameOnly(List[Index1]));
    //b := pnat_char(ExtractFileNameOnly(List[Index2]));
    Result := strnatcasecmp(a, b)

    //if List.CaseSensitive then
    // Result := strnatcmp(a, b)
    //else
    // Result := strnatcasecmp(a, b);

    //Result := -AnsiStrlComp(PChar(Node1.Text), PChar(Node2.Text));
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    //RootPath:=ExtractFilePath(Application.ExeName) + 'TestData';
    RootPath:='D:\';
    Memo1.Clear;
    TreeView1.Items.Clear;
    DirToTreeView(TreeView1, RootPath, nil, true, '*');

    list := TStringList.Create;
    EnumText(RootPath, TreeView1.Items.GetFirstNode);
    Memo1.text := list.text;

    // 对 list 排序
    //list.CustomSort(@CompareStr); // 对文件名列表排序
    memo1.Append('-----');

```

```

memo1.Append(list.text );

SetIcons(TreeView1, list);

//TreeView1.CustomSort(@MyTreeViewSort); //CustomSort(@MyTreeViewSort);
TreeView1.CustomSort(@TreeviewAlphaSort);

//list.Free;
end;

procedure TForm1.FormDestroy(Sender: TObject);
begin
    list.Free;
end;
end.

```

转自：lazarus：对 treeview 控件内容进行自然排序_老狼 8848 的博客-CSDN 博客

```

procedure TForm1.Button4Click(Sender: TObject);
var
    a, b: pnat_char;
    i, j: integer;
    s: string;
begin
    //使用冒泡法对 Memo1 的行进行排序
    for i:=0 to Memo1.Lines.Count-1 do
    begin
        for j:=0 to Memo1.Lines.Count-2 do
        begin
            a:=pnat_char(Memo1.Lines[j]);
            b:=pnat_char(Memo1.Lines[j+1]);
            if strnatcasecmp(a,b)>0 then
            begin
                s:=Memo1.Lines[j];
                Memo1.Lines[j]:=Memo1.Lines[j+1];
                Memo1.Lines[j+1]:=s;
            end;
        end;
    end;
end;

```

排序前：

```

1
2.1
1.1.1
1.1.2
1.0.0
2.0.0
2.1.2
2.1.0

```

排序后：

1
1.0.0
1.1.1
1.1.2
2.0.0
2.1
2.1.0
2.1.2

● 7、自行编译的 CudaText 在 linux for GTK2 下不能输入中文

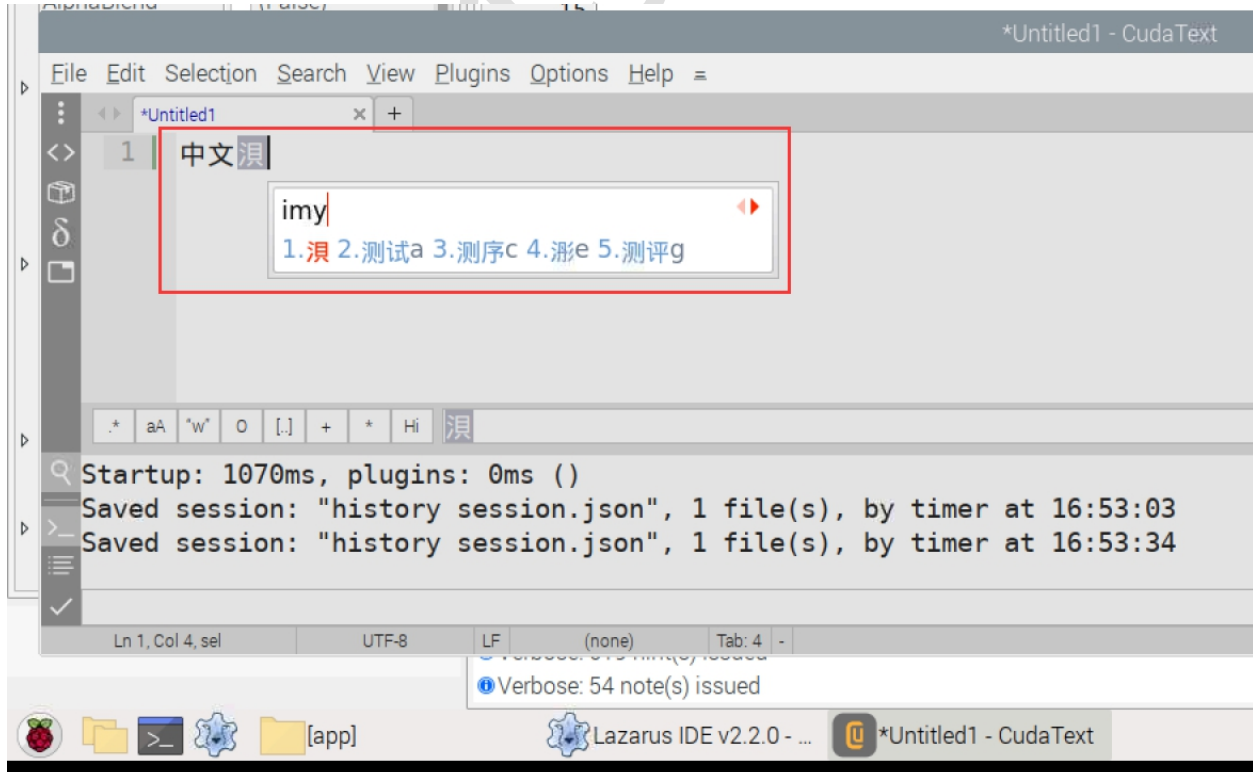
遇到一件奇怪的事，从 CudaText 官网下载编译好的 CudaText 在 linux 下能输入中文，但自行编译的 CudaText 不能输入 中文。从 atsynedit.pas 发现需要 lazarus2.3 版本才可以启用 GTK2_IME_CODE，为了在 lazarus 2.2.0 上实现中文输入，只需将 atsynedit.pas 66 行开始：

```
{IFDEF LCLGTK2}  
  {IF (LCL_FULLVERSION >= 2030000)}  
    {DEFINE GTK2_IME_CODE}  
  {ENDIF}{ENDIF}
```

改为：

```
{IFDEF LCLGTK2}{DEFINE GTK2_IME_CODE}{ENDIF}
```

然后重新编译 CudaText 就可以正常输入中文。



● 8、lazarus 填坑小记

使用以下方法可以解决 **lazarus** 在 **linux** 日期格式引起 **unidac** 日期字段保存出错：

```
fs:TFormatSettings;  
begin  
  fs.DateSeparator:='-';  
  fs.TimeSeparator:=':';  
  fs.ShortDateFormat:='yyyy-mm-dd';  
  fs.LongDateFormat:='yyyy-mm-dd';  
  fs.LongTimeFormat:='hh:NN:ss';  
  fs.ShortTimeFormat:='hh:NN:ss';  
  DefaultFormatSettings:=fs;  
end;
```

发现原来使用正常的 **float** 字段只显示整数

1	15.8
---	------

平均强度	测区强度
24.0	15.8
33.0	29.3
22.0	13.6
30.0	24.3
34.0	31.0
33.0	29.8
40.0	42.5
44.0	51.1
32.1	27.7
45.0	53.4

按上面的方法设置日期格式后 **float** 字段变为整数：

1	15 8
---	------

平均强度	测区强度
24	15
33	29
22	13
30	24
34	31
33	29
40	42
44	51
32	27
45	53

后发现日期格式按以下方面设置后日期和 **float** 在 **windows** 和 **Linux** 都正常。

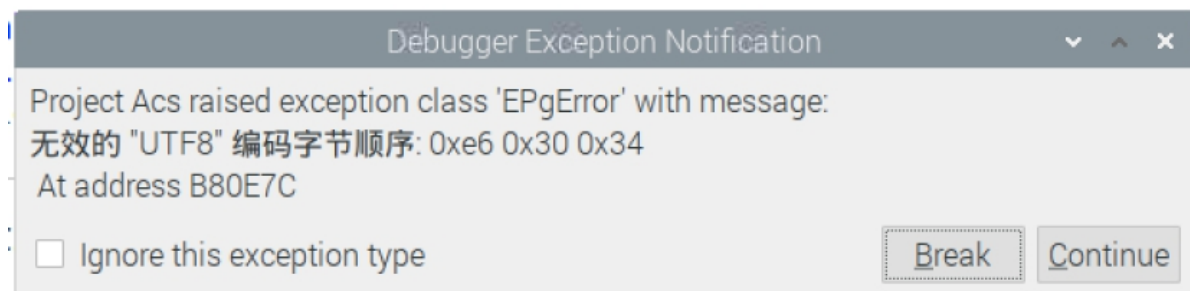
```
DefaultFormatSettings.ShortTimeFormat:='yyyy-mm-dd';  
DefaultFormatSettings.ShortTimeFormat:='hh:NN:ss';  
DefaultFormatSettings.LongDateFormat:='yyyy-mm-dd';  
DefaultFormatSettings.LongTimeFormat:='hh:NN:ss';  
DefaultFormatSettings.DateSeparator:='-';  
DefaultFormatSettings.TimeSeparator:=':';
```

- **9、lazarus 编译的应用由于 linux 日期格式引起 unidac 日期字段保存时出错的问题 (2022.04.28 修正)**

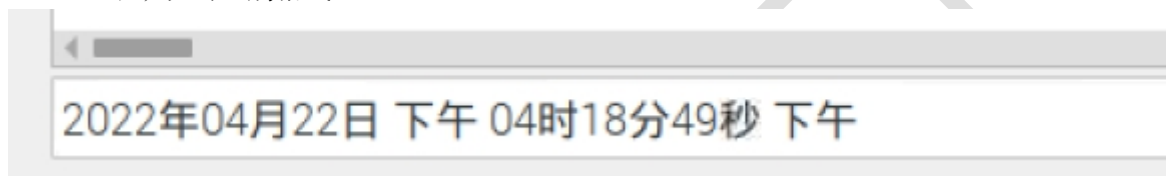
lazarus 生成的程序在 linux arm64(银河麒麟和 raspberry)使用 unidac 控件(数据库 postgresql), 采用以下代码保存日期字段时, 提示“无效 UTF8 的编码字节顺序”, 百思不解这奇怪的问题, 后来发现 DateTimeToStr 转换出现的日期格式和 windows 的不一样。

```
procedure TForm1.Button2Click(Sender: TObject);
begin
    UniQuery1.Edit;
    UniQuery1.FieldByName('prdate').AsDateTime:=now;
    UniQuery1.Post;
end;
```

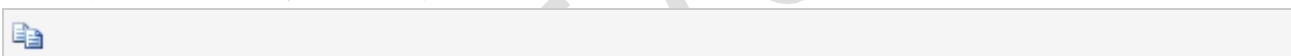
post 时提示出错:



Linux 下默认日期格式:



尝试将 lazarus 的日期格式改为与 windows 的一样:



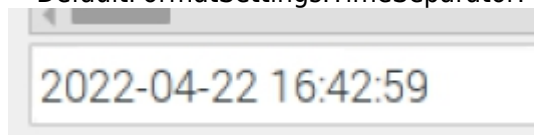
```
var fs:TFormatSettings;begin
```

```
fs.DateSeparator:='-';
fs.TimeSeparator:=':';
fs.ShortDateFormat:='yyyy-mm-dd';
fs.LongDateFormat:='yyyy-mm-dd';
fs.LongTimeFormat:='hh:NN:ss';
fs.ShortTimeFormat:='hh:NN:ss';
DefaultFormatSettings:=fs;end;
```

//2022-04-28 发现使用以上方法在 windows 下 float 字段会只显示整数的 Bug

//使用以下方法设置后日期、float 在 windows、linux 都正常 2022.04.28

```
DefaultFormatSettings.ShortTimeFormat:='yyyy-mm-dd';
DefaultFormatSettings.ShortTimeFormat:='hh:NN:ss';
DefaultFormatSettings.LongDateFormat:='yyyy-mm-dd';
DefaultFormatSettings.LongTimeFormat:='hh:NN:ss';
DefaultFormatSettings.DateSeparator:='-';
DefaultFormatSettings.TimeSeparator:=':';
```



改变日期格式后, unidac 日期字段保存也正常了, 说明我遇到的问题是日期格式引起的, 如你使用过程也遇到同样问题可尝试使用这个方法说不定能解决。

还有另一种解决方法就是采用 SQL Update, 也能正常更新日期字段:

```
UniQuery1.SQL.Text:='update power set prdate='+chr(39)+lrdatetimeostr(now)+chr(39)+' where username="1"';
UniQuery1.ExecSQL;
```

● 10、Lazarus Delphi mode 和 OBJFPC mode 的区别

Delphi mode

- 1、你不能在函数指针或者运算符上使用地址符号
- 2、事先的申明可以省略参数
- 3、AnsiString 是默认的,
- 4、同名函数是不允许的
- 5、备注嵌套是不允许的
- 6、objpas 加载在 system 之后, 明显的区别在于 integer 是 longint
- 7、类方法的参数可以与类属性同名

OBJFPC mode

- 1、你必须在函数指针或者运算符上加上地址符号
- 2、事先的生命必须完全和后面的实现相符合, 相反, 你可以在实现中省略参数
- 3、同名函数是允许的
- 4、备注嵌套是允许的
- 5、objpas 加载在 system 之后, (和 delphi 一样)
- 6、你可以使用 cvar 类型
- 7、PChars 被自动当作 strings 处理
- 8、类方法的参数不能与类属性同名
- 9、String 默认是 ShortString,

● 11、linux 使用 lazarus 源码安装 lazarus 及交叉编译环境(windows x86_64)

以 linux ARM64 环境为例, 到官网下载与 CPU 适配的 FPC 执行文件、FPC 源码和 Lazarus 源码(整合安装包已修复 lazarus 在 linux 代码编辑不能输入中文的 Bug,整合安装包下载链接:

<https://pan.baidu.com/s/1LsHv5COD6up5PJCBLL-DZBw?pwd=tpm6> 提取码: tpm6), 并解压到适当的目录:

lazarus 源码:

/home/用户名/lazarus

fpc 执行文件:

/home/用户名/lazarus/fpc-3.2.2.aarch64-linux

fpc 源码:

/home/用户名/lazarus/fpcsrc

整合包的版本为:

Lazarus 2.2.0, FPC 3.2.2

一、安装 FPC

cd fpc-3.2.2.aarch64-linux

sudo ./install.sh

提示下列时, 输入? 后的内容

Install prefix (/usr or /usr/local)? 直接回车

Install textmode IDE(y/n)?y

Install FCL(y/n)?y

Install Packages(y/n)?y

Install Documentation (y/n)?y

Install Demo (y/n)?y

二、安装 lazarus

如果是第一次安装 lazarus, 需安装以下软件:

```
sudo apt-get install gdb gcc libx11-dev libgdk-pixbuf2.0-dev libcairo2-dev lpango-1.0
libpangox-1.0-dev xorg-dev libgtk2.0-dev libgdk-pixbuf2.0-dev libpango1.0-dev
cd lazarus
```

```
make clean all
```

等安装完成就可以。

三、编译“交叉编译”程序

```
sudo -i export FPCVER="3.2.2"
```

```
cd /home/用户名/lazarus/fpc-3.2.2/
```

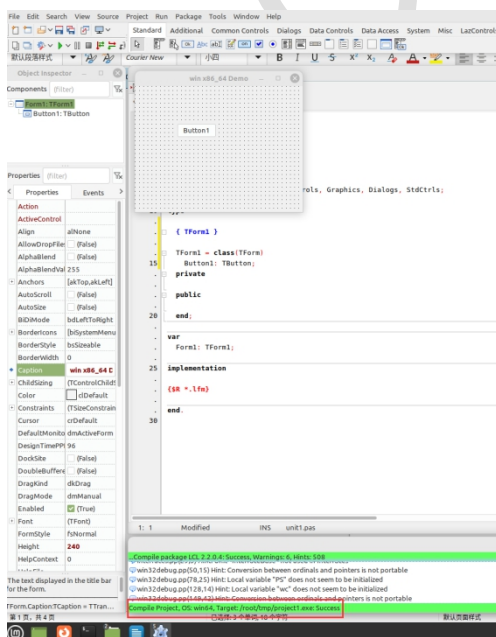
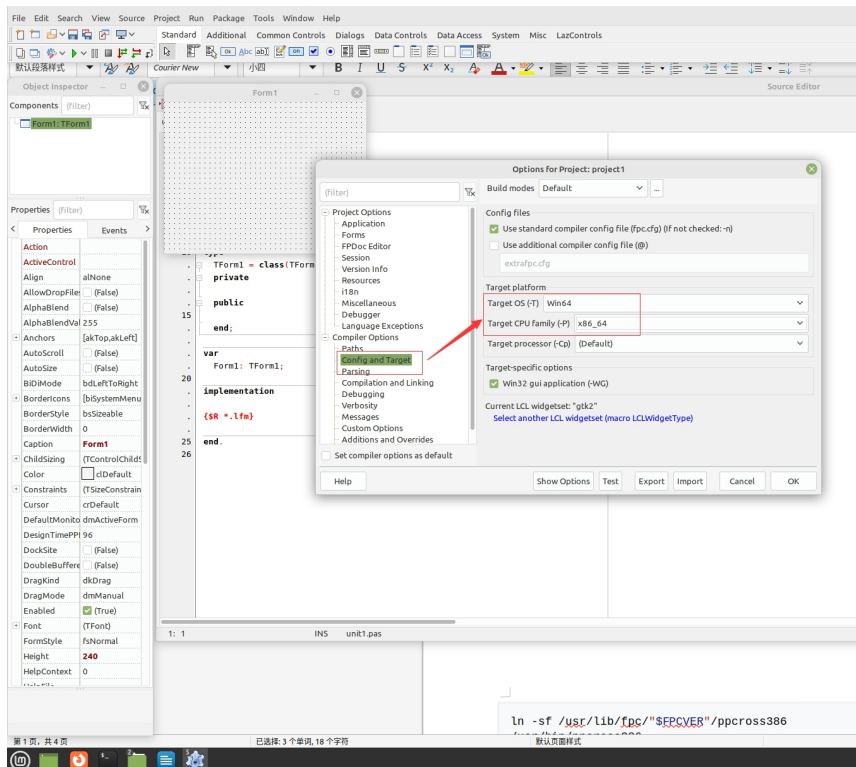
```
make clean all OS_TARGET=win64 CPU_TARGET=x86_64
```

建符号连接

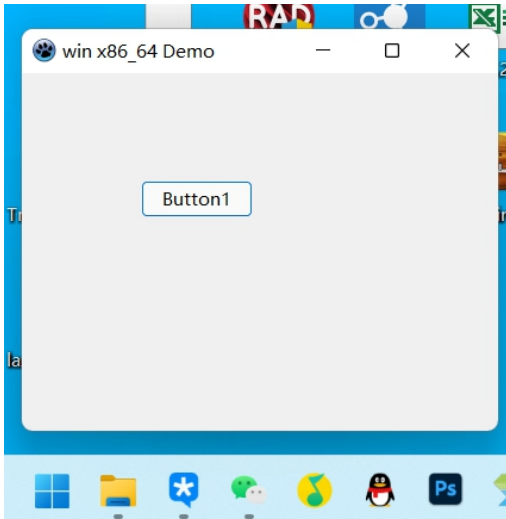
```
make crossinstall OS_TARGET=win64 CPU_TARGET=x86_64 INSTALL_PREFIX=/usr
```

```
ln -sf /usr/lib/fpc/"$FPCVER"/ppcrossx64 /usr/bin/ppcrossx64
```

完成交叉编译环境后就可以在 linux 编译 win64 x86_64 的应用程序



编译成功的程序：



附录：

1、最新 lazarus 源码：

https://sourceforge.net/projects/lazarus/files/Lazarus%20Zip%20_%20GZip/

2、FPC 执行文件：

<https://sourceforge.net/projects/freepascal/files/Linux/3.2.2/>

3、fpc 源码：

<https://sourceforge.net/projects/freepascal/files/Source/3.2.2/>

● 12、lazarus 使用技巧 2

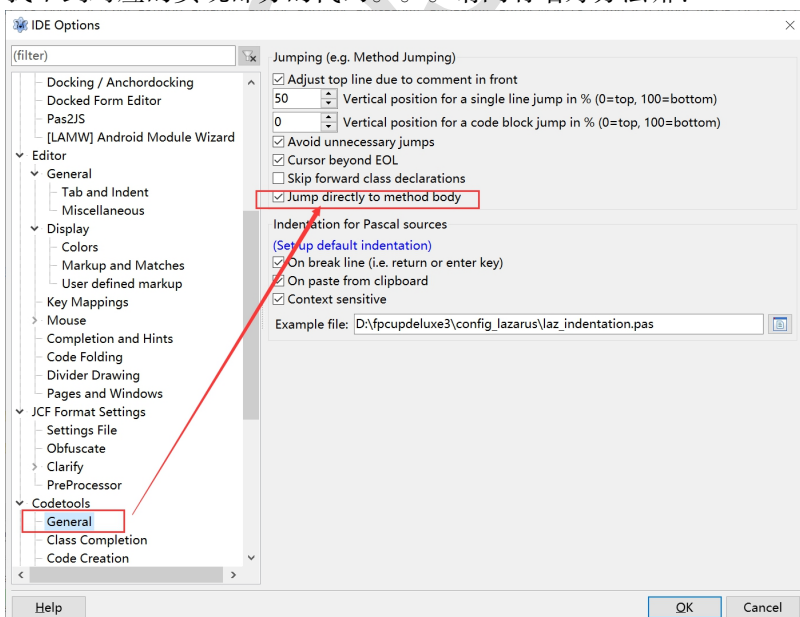
lazarus 以下 2 个小技巧跨平台也能使用：

1、根据平台自动选择适合的路径分隔符（\或/）

`S:=AppendPathDelim(ExtractFileDir(ParamStr(0))) + 'aaa1.pdf';`

2、使用平台默认的程序打开文档： `OpenDocument(S);`

3、新手求教个 lazarus IDE 编辑器的问题，由于 FPC 的源代码很多都用了 `{I XXX}` 的方式，将一个 pas 文件拆分成了多个文件，导致在 IDE 中按 `Ctrl+` 鼠标左键跟踪函数或属性时，自动跳到了申明部分，但是找不到对应的实现部分的代码。。。请问有啥好办法嘛？



● 13、lazarus linux 下使用 powerPDF 控件中文乱码及英文和中文等宽的解决方法

lazarus linux 下使用 powerPDF 中文乱码按网上的修改方法还是存在出现乱码问题，经跟踪 powerpdf 源码，除启用 PRreort.pas {`#DEFINE USE_GBFFONTS`}外，还要修改 pdfdoc.pas TPdfCanvas.ShowText(const s: string),并添加 LConvEncoding 单元,使用 UTF8ToCP936 将 UTF8 转为 CP936。

```
pdfdoc.pas:
procedure TPdfCanvas.ShowText(const s: string);
var
    FString: string;
begin
    if _HasMultiByteString(s) then
        FString := '<' + _StrToHex(s) + '>';
    else
        FString := '(' + _EscapeText(s) + ')';
    WriteString(FString + ' Tj'#10);
```

```
procedure TPdfCanvas.ShowText(const s: string);
var
    FString: string;
begin
    FString := '<' + _StrToHex(LConvEncoding.UTF8ToCP936(s,true)) + '>';
    WriteString(FString + 'Tj'#10);
end;
```

按上面的方法就可以解决中文乱码的问题，但存在英文和中文字体等宽的问题，要解决英文宽度和中文等宽的问题还需在 pdfGBFonts.pas 增加英文字符宽度定义及 procedure TPdfGBFixedFont.AddDescendantFontItem(ADescendantFont: TPdfDictionary)修改为：

const

修改后的 powerpdf 在 windows 和 linux 都可以正确保存含中文的 pdf，也解决了英文宽度和中文等宽的问题。

中文 a b c
中文 深圳 1 2 3 a b c
abc1234

解决英文与中文等宽后的 PDF:

中文abc , . ! @ #
abc123
ABC123
安全

修改后的 PowerPDF 下载链接:

<https://pan.baidu.com/s/1UzmzyAB1mqegyzUasSiLtg>

提取码: vvqn

● 14、定制 **lazarus compiler config**

lazarus 默认的 Target OS 和 Target CPU 包含很多用不上的 OS 和 CPU, 我平时只用到 Linux、windows 和 Android, CPU 为 aarch64、arm、i386 和 x86_64, 为方便选择不同的 OS 和 CPU, 只需修改

lazarus\components\codetools\definetemplates.pas

FPCOperatingSystemNames: **array**[1..38] of shortstring = (

```
'linux',  
'win32','win64','wince',  
'darwin','macos',  
'freebsd','netbsd','openbsd','dragonfly',  
'aix',  
'amiga',  
'android',  
'aros',  
'atari',  
'beos',  
'embedded',  
'emx',  
'freertos',  
'gba',
```

```
'go32v2',  
'haiku',  
'iphonesim',  
'ios',  
'java',  
'msdos',  
'morphos',  
'nds',  
'netware',  
'netwlibc',  
'os2',  
'palmos',  
'qnx',  
'solaris',  
'symbian',  
'watcom',  
'wdosx',  
'wii'
```

```
);
```

```
FPCOperatingSystemCaptions: array[1..38] of shortstring =(  
'AIX',  
'Amiga',  
'Android',  
'AROS',  
'Atari',  
'BeOS',  
'Darwin',  
'DragonFly',  
'Embedded',  
'emx',  
'FreeBSD',  
'FreeRTOS',  
'GBA',  
'Go32v2',  
'Haiku',  
'iPhoneSim',  
'iOS',  
'Java',  
'Linux',  
'MacOS',  
'MorphOS',  
'MSDOS',  
'NDS',  
'NetBSD',  
'NetWare',  
'NetwLibC',  
'OpenBSD',  
'OS2',  
'PalmOS',  
'QNX',  
'Solaris',  
'Symbian',  
'Watcom',  
'wdosx',  
'Win32',  
'Win64',  
'WinCE',  
'Wii'
```

```
);
```

改为:

```
FPCOperatingSystemNames: array[1..4] of shortstring =(  
'linux',  
'win32',  
'win64',  
'android'
```

```
);
```

```
FPCOperatingSystemCaptions: array[1..4] of shortstring =(  
'Linux',  
'Win32',  
'Win64',
```

```
'Android'  
);
```

2、将 FPCProcessorNames

FPCProcessorNames: `array[1..14] of shortstring` =(

```
'aarch64',  
'arm',  
'avr',  
'i386',  
'i8086',  
'jvm',  
'm68k',  
'mips',  
'mipsel',  
'powerpc',  
'powerpc64',  
'sparc',  
'x86_64',  
'xtensa'
```

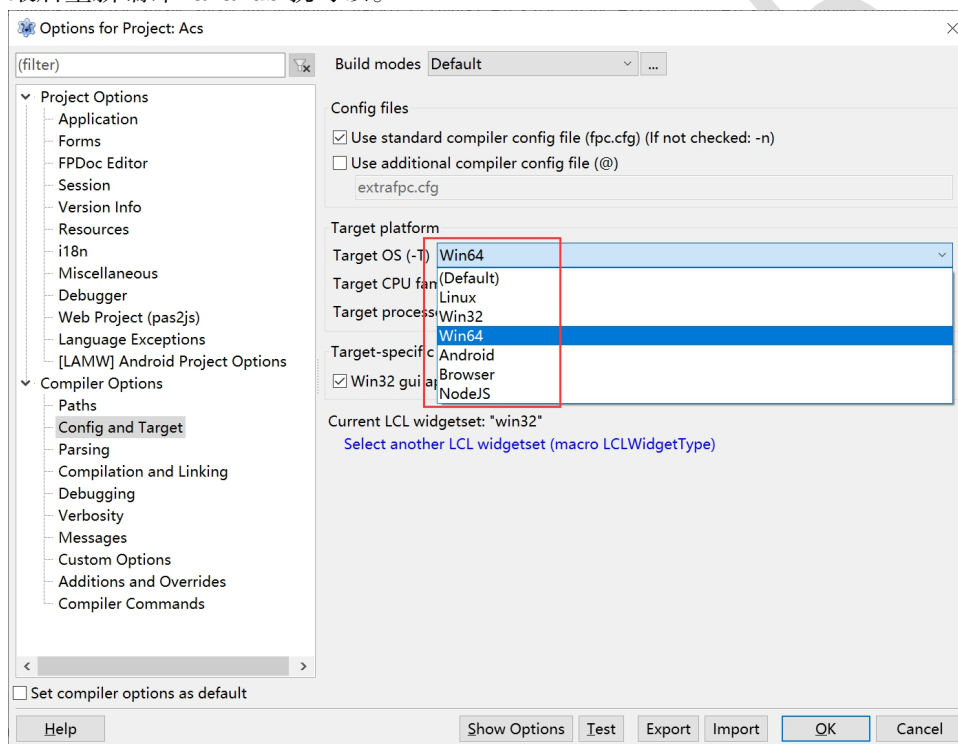
```
);
```

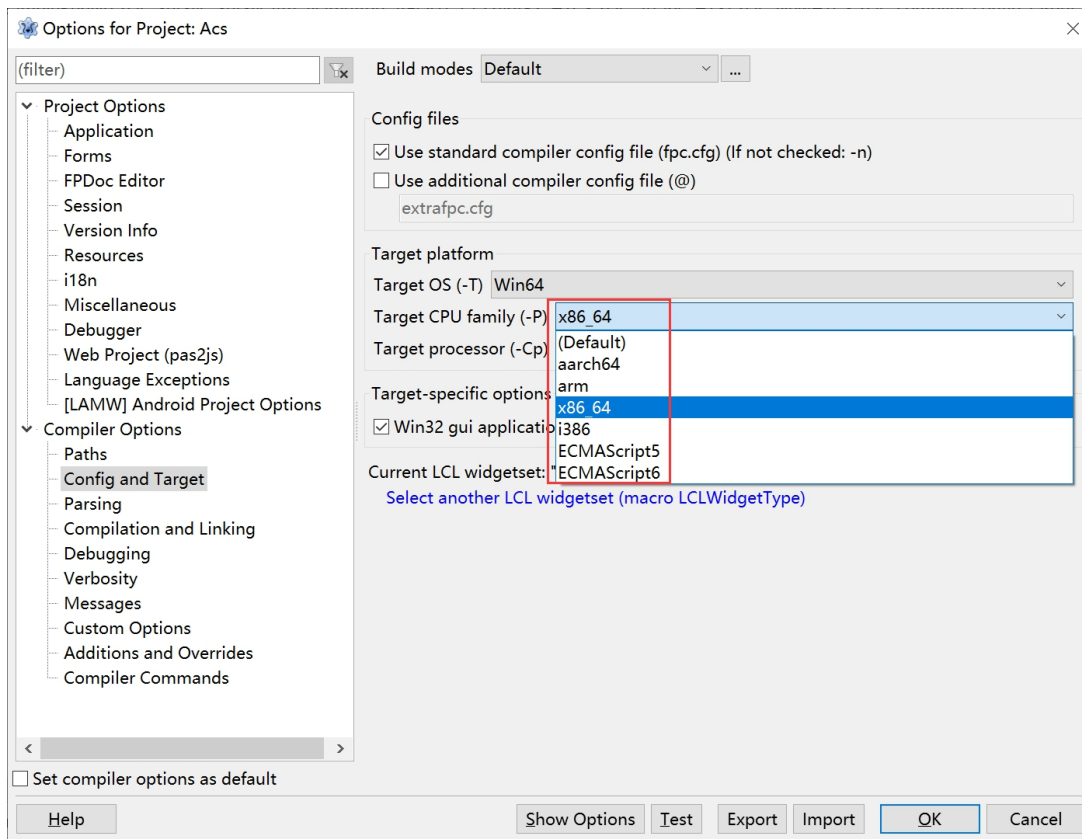
改为:

```
FPCProcessorNames: array[1..4] of shortstring =(  
'aarch64',  
'arm',  
'x86_64',  
'i386'
```

```
);
```

最后重新编译 lazarus 就可以。





● 15、Lazarus+LAMW 强制 APP 屏幕方向

在 OnJNIPrompt、OnRotate 按下面的添加就可以强制指定横屏或竖屏了。

procedure TAndroidModule1.AndroidModule1JNIPrompt(Sender: TObject);

begin

 Self.SetScreenOrientationStyle(ssPortrait); //强制竖屏

end;

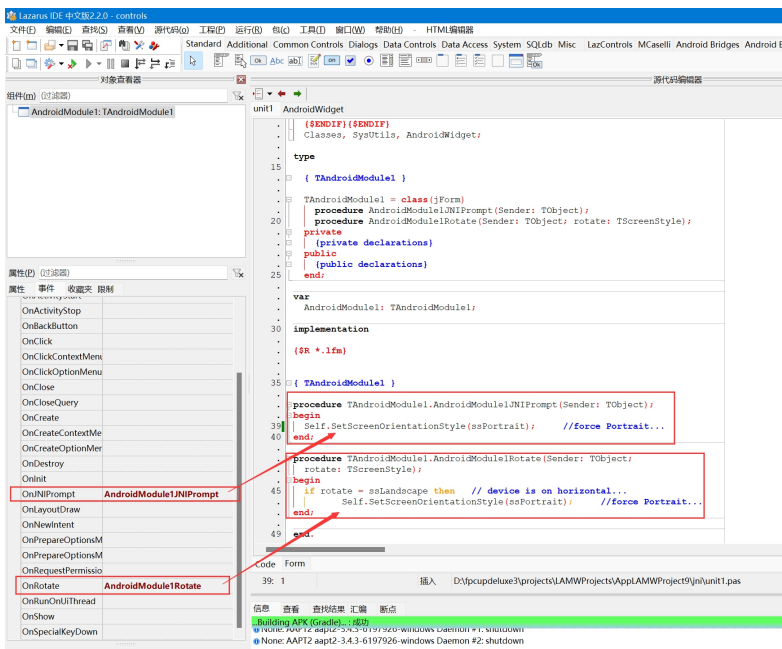
procedure TAndroidModule1.AndroidModule1Rotate(Sender: TObject;
 rotate: TScreenStyle);

begin

if rotate = ssLandscape **then** // device is on horizontal...

 Self.SetScreenOrientationStyle(ssPortrait); //强制竖屏

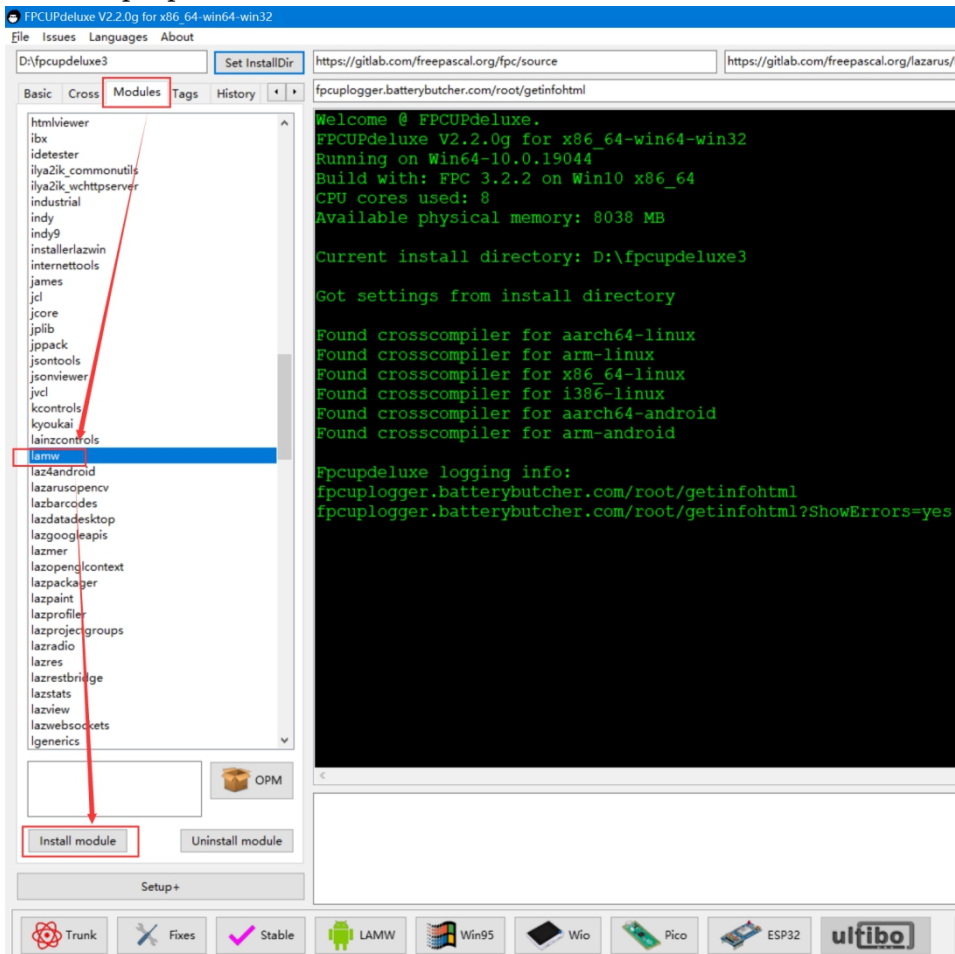
end;



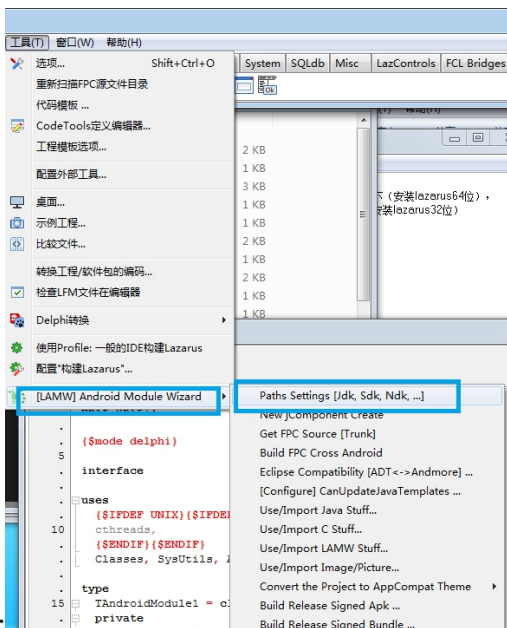
● 16、lazarus LAMW 使用简要说明

一、安装和配置 LAMW

1、使用 fpcupdeluxe 安装 LAMW:

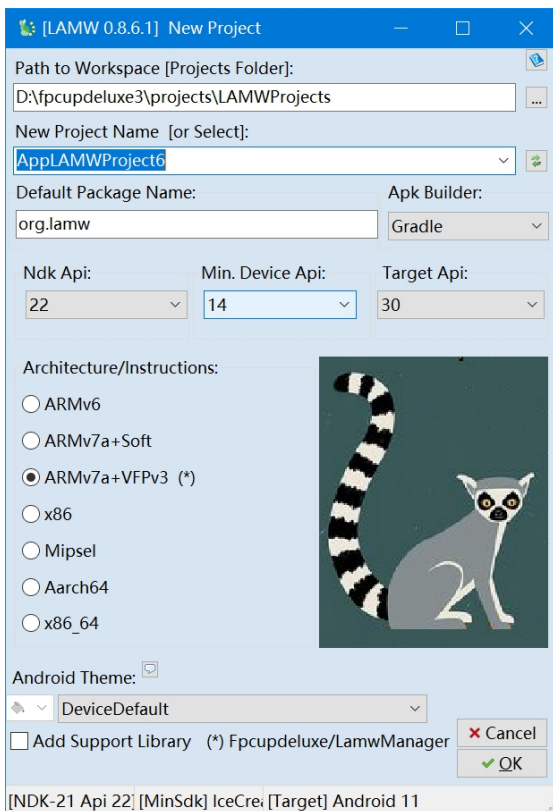


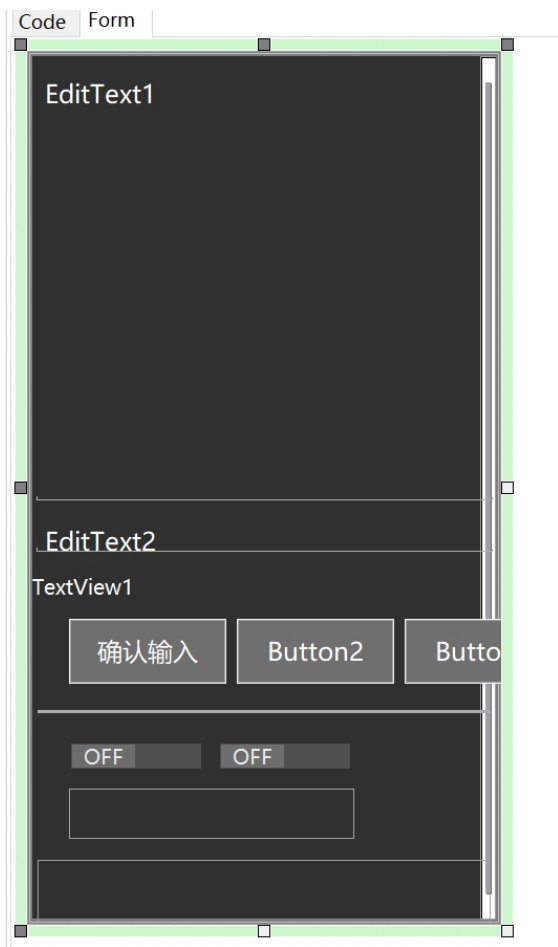
2、配置 lazarus/LAMW



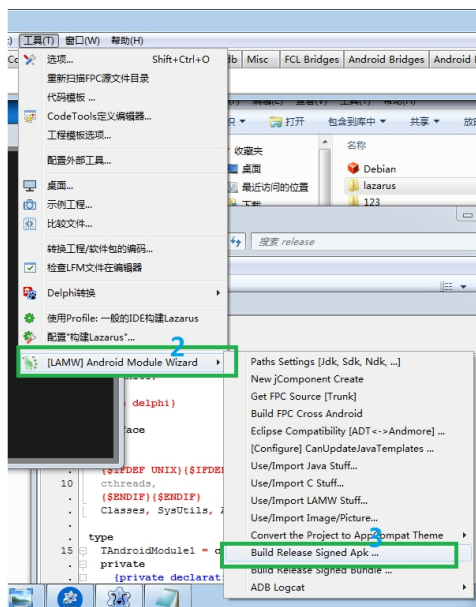
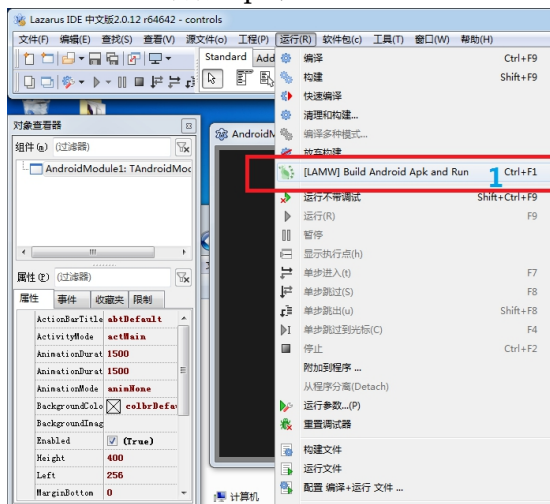
二、新建安卓 APP

1、创建新工程：





2、编译和制作 apk，见图：



如果不执行图中 2 和 3 进行签名则安装时会出现错误，如不能生成带签名的 APK，请参考修复 [lazarus LAMW release-keystore.bat](#) 无法生成.keystore 文件。

3、制作并签名后的 apk 位置，见图：

LAMWProjects > AppLAMWProject5 > build > outputs > apk > release					搜索"release"
名称	修改日期	类型	大小		
AppLAMWProject5-armeabi-v7a-release-unsigned.apk	2022-01-04 09:10	APK 文件	250 KB		
AppLAMWProject5-armeabi-v7a-release-unsigned-aligned.apk	2022-01-04 09:11	APK 文件	250 KB		
AppLAMWProject5-release.apk	2022-01-04 09:20	APK 文件	257 KB		
AppLAMWProject5-release.apk.idsig	2022-01-04 09:20	IDSIG 文件	6 KB		
output.json	2022-01-04 09:10	JSON 文件	1 KB		

● 17、lazarus LAMW 开发安卓 APP 时，GUI 如何布局？

lazarus LAMW 开发安卓 APP 时，GUI 如何布局？

可视组件请配置这些属性：

Anchor

PosRelativeToParent

PosRelativeToAnchor

LayoutParamHeight

LayoutParamWhidth

例

1 - 在 LAMW Form 上放置一个 **jTextView** 组件并设置属性：

PosRelativeToParent

rpCenterHorizontal = [True]

rpTop = [True]

Text = GUI Design WYSIWYG

(Because only Position Relative to Parent(form) is need to localize jTextView1)

2 - 在 LAMW Form 上放置一个 **jEditText** 组件并设置属性：

Anchor = jTextView1

LayoutParamWidth = lpTwoThirdOfParent

PosRelativeToAnchor

raBelow = [True]

PosRelativeToParent

rpCenterHorizontal = [True]

Text = Ok

(Because we want it to be bellow the jTextView1)

3 - 将 **jButton** 组件放在 LAMW 窗体上并设置属性：

Anchor = jEditText1

LayoutParamWidth = lpTwoThirdOfParent

PosRelativeToAnchor

raBelow = [True]

PosRelativeToParent

rpCenterHorizontal = [True]

Text = Sample

(Because we want it to be bellow the jEditText1)

4 - 提示：

要更改可视组件的宽度/高度，应配置 **LayoutParamWhidth** 和 **LayoutParamHeight**！

● 18、lazarus 编写的软件（windows 和 Linux）只运行一个实例的方法

本方法参考并使用了 CudaText 的 2 个单元（appuniqueinstance.pas 和 appuniqueinstancebase.pas），appuniqueinstance.pas 增加 RunOnce。
使用方法：

在项目文件 uses 添加 AppUniqueInstance 单元，Application.Initialize 下添加 runonce（将红色字体的内容添加到你的工程文件[lpr]）就可以实现在 windows 和 linux 只运行一个实例。

```
program project1;
{$mode objfpc}{$H+}
uses
  {$IFDEF UNIX}
  cthreads,
  {$ENDIF}
  {$IFDEF HASAMIGA}
  athreads,
  {$ENDIF} AppUniqueInstance, Interfaces, // this includes the LCL widgetset  Forms, Unit1;{$R *.res}
begin
  RequireDerivedFormResource:=True;
  Application.Scaled:=True;
  Application.Initialize;
  RunOnce;
  Application.CreateForm(TForm1, Form1); Application.Run;
end.
```

修改后的 AppUniqueInstance，增加 RunOnce 使用更简单。

```
unit AppUniqueInstance;
{
  UniqueInstance is a component to allow only a instance by program

  Copyright (C) 2006 Luiz Americo Pereira Camara
  pascalive@bol.com.br

  This library is free software; you can redistribute it and/or modify it
  under the terms of the GNU Library General Public License as published by
  the Free Software Foundation; either version 2 of the License, or (at your
  option) any later version with the following modification:

  As a special exception, the copyright holders of this library give you
  permission to link this library with independent modules to produce an
  executable, regardless of the license terms of these independent modules, and
  to copy and distribute the resulting executable under terms of your choice,
  provided that you also meet, for each linked independent module, the terms
  and conditions of the license of that module. An independent module is a
  module which is not derived from or based on this library. If you modify
  this library, you may extend this exception to your version of the library,
  but you are not obligated to do so. If you do not wish to do so, delete this
  exception statement from your version.

  This program is distributed in the hope that it will be useful, but WITHOUT
  ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
  FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License
  for more details.

  You should have received a copy of the GNU Library General Public License
  along with this library; if not, write to the Free Software Foundation,
  Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.}

```

```
{$mode objfpc}{$H+}{$if not defined(Windows) or (FPC_FULLVERSION >= 30001)}{$define
PollIPCMessage}{$endif}
```

```
interface
```

```
uses
```

```
Forms, Classes, SysUtils, simpleipc, ExtCtrls;
```

```
type
```

```
TOnOtherInstance = procedure (Sender : TObject; ParamCount: Integer; const Parameters: array of String) of object;
```

```
{ TUniqueInstance }
```

```
TUniqueInstance = class(TComponent)
```

```

private
  FIdentifier: String;
  FOnOtherInstance: TOnOtherInstance;
  FUpdateInterval: Cardinal;
  FEnabled: Boolean;
  FPriorInstanceRunning: Boolean;
  {$ifdef PollIPCMessage}
  Timer: TTimer;
  {$endif}
  procedure ReceiveMessage(Sender: TObject);
  {$ifdef PollIPCMessage}
  procedure CheckMessage(Sender: TObject);
  {$endif}
public
  constructor Create(AOwner: TComponent); override;
  destructor Destroy; override;
  property PriorInstanceRunning: Boolean read FPriorInstanceRunning;
  procedure Loaded(const AParams: array of string); reintroduce; //Alexey
published
  property Enabled: Boolean read FEnabled write FEnabled default False;
  property Identifier: String read FIdentifier write FIdentifier;
  property UpdateInterval: Cardinal read FUpdateInterval write FUpdateInterval default 1000;
  property OnOtherInstance: TOnOtherInstance read FOnOtherInstance write FOnOtherInstance;
end;
procedure RunOnce;
implementation
uses
  StrUtils, AppUniqueInstanceBase;
{ TUniqueInstance }
procedure RunOnce;
var
  AppUniqInst: TUniqueInstance = nil;
  CmdParams: array of string;
begin
  if not Assigned(AppUniqInst) then
    AppUniqInst := TUniqueInstance.Create(nil);
  if not AppUniqInst.Enabled then
    begin
      AppUniqInst.Enabled := true;
      AppUniqInst.Loaded(CmdParams);
      if AppUniqInst.PriorInstanceRunning then
        begin
          AppUniqInst.free;
          Application.Terminate;
        end;
    end
  else
    begin
      if Assigned(AppUniqInst) then
        AppUniqInst.free;
    end;
end;
end;
procedure TUniqueInstance.ReceiveMessage(Sender: TObject);
var
  ParamsArray: array of String;
  Params: String;
  Count, i: Integer;begin
  if Assigned(FOnOtherInstance) then
  begin
    //MsgType stores ParamCount
    Count := FIPCServer.MsgType;
    SetLength(ParamsArray, Count);
    Params := FIPCServer.StringMessage;
    for i := 1 to Count do
      ParamsArray[i - 1] := ExtractWord(i, Params, [ParamsSeparator]);
    FOnOtherInstance(Self, Count, ParamsArray);
  end;
end;
{$ifdef PollIPCMessage}
procedure TUniqueInstance.CheckMessage(Sender: TObject);

```

```

begin
    if FIPCTServer.Active then
        FIPCTServer.PeekMessage(1, True);
    end;
{$endif}
procedure TUniqueInstance.Loaded(const AParams: array of string);
var
    IPCClient: TSimpleIPCClient;
begin
    if not (csDesigning in ComponentState) and FEnabled then
        begin
            IPCClient := TSimpleIPCClient.Create(Self);
            IPCClient.ServerId := GetServerId(FIdentifier);
            if not Assigned(FIPCTServer) and IPCClient.ServerRunning then
                begin
                    //A older instance is running.
                    FPriorInstanceRunning := True;
                    //A instance is already running
                    //Send a message and then exit
                    if Assigned(FOnOtherInstance) then
                        begin
                            IPCClient.Active := True;
                            IPCClient.SendStringMessage(ParamCount, GetFormattedParams(AParams));
                        end;
                    Application.ShowMainForm := False;
                    Application.Terminate;
                end
            else
                begin
                    if not Assigned(FIPCTServer) then
                        InitializeUniqueServer(IPCClient.ServerId);
                    FIPCTServer.OnMessage := @ReceiveMessage;
                    //there's no more need for IPCClient
                    IPCClient.Destroy;
                    {$ifdef PollIPCMessage}
                    if Assigned(FOnOtherInstance) and (Timer=nil) then
                        begin
                            Timer := TTimer.Create(Self);
                            Timer.Interval := FUpdateInterval;
                            Timer.OnTimer := @CheckMessage;
                        end;
                    {$endif}
                end;
            end;
        end;
    inherited Loaded;
end;
constructor TUniqueInstance.Create(AOwner: TComponent);
begin
    inherited Create(AOwner);
    FUpdateInterval := 1000;
end;
destructor TUniqueInstance.Destroy;
begin
    if Assigned(Timer) then
        begin
            Timer.Enabled := False;
            Timer.OnTimer := nil;
            FreeAndNil(Timer);
        end;
    inherited Destroy;
end;
end.

unit AppUniqueInstanceBase;
{$mode objfpc}{$H+}
interface
uses
    Classes, SysUtils, simpleipc;
const
    ParamsSeparator = #13;

```

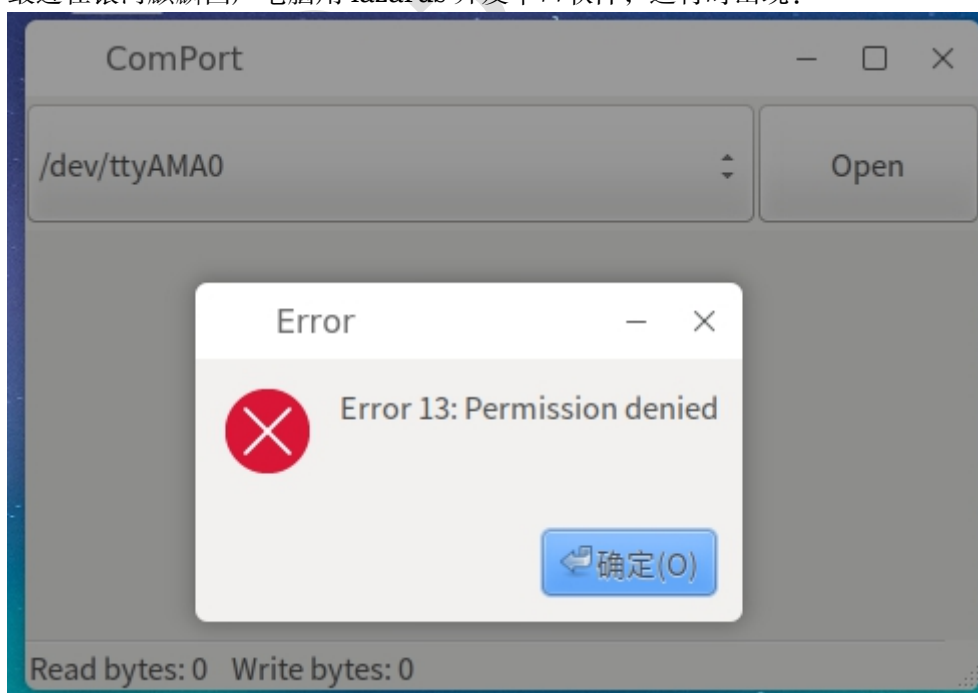
```

var
  FIPCTServer: TSimpleIPCTServer;
procedure InitializeUniqueServer(const ServerId: String);
function GetFormattedParams(const AParams: array of string): String; //Alexey
function GetServerId(const Identifier: String): String;
implementation
uses
  LazUTF8;
const
  BaseServerId = 'tuniqueinstance_';
procedure InitializeUniqueServer(const ServerId: String);
begin
  //It's the first instance. Init the server
  if FIPCTServer = nil then
  begin
    FIPCTServer := TSimpleIPCTServer.Create(nil);
    FIPCTServer.ServerID := ServerId;
    FIPCTServer.Global := True;
    FIPCTServer.StartServer;
  end;
end;
function GetFormattedParams(const AParams: array of string): String; //Alexey
var
  i: Integer;
begin
  Result := '';
  for i := Low(AParams) to High(AParams) do
    Result := Result + AParams[i] + ParamsSeparator;
  end;
function GetServerId(const Identifier: String): String;
begin
  if Identifier <> '' then
    Result := BaseServerId + Identifier
  else
    Result := BaseServerId + ExtractFileName(ParamStrUTF8(0));
  end;
finalization
  FIPCTServer.Free;
end.

```

● 19、添加 linux 串口访问权限

最近在银河麒麟国产电脑用 **lazarus** 开发串口软件，运行时出现：



提示权限不够，是因为用户没有 **/dev/tty** 的访问权限，只需将用户添加到 **dialout** 组就可以解决权限不够的

问题。

解决方法如下：

1.由于 `tty` 属于“`dialout`”组，比如用户名是 `qilin`，
用命令查看用户所属的组别

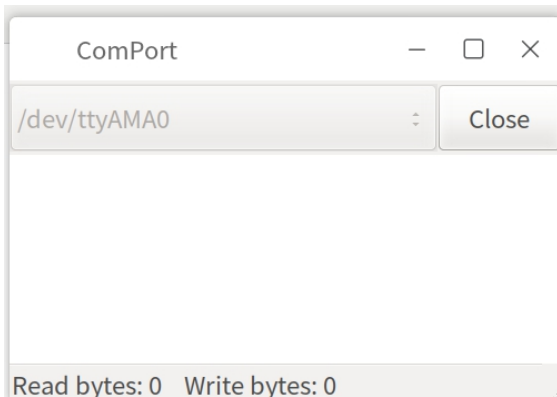
`groups qilin`

2.如果没有隶属“`dialout`”，则把该用户加入进去

`sudo gpasswd --add qilin dialout`

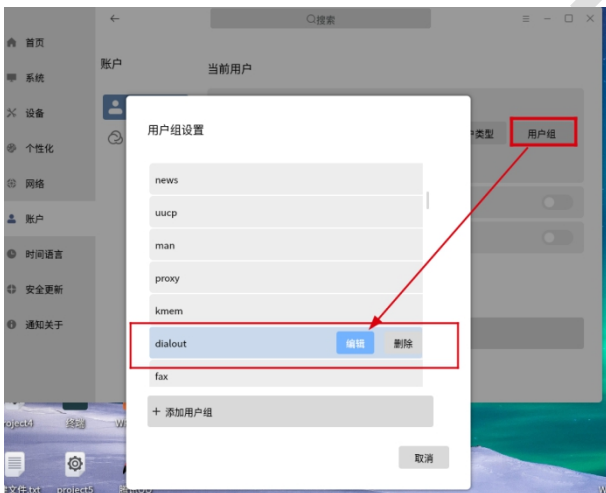
3.重新启动后，再登录系统就可以了。

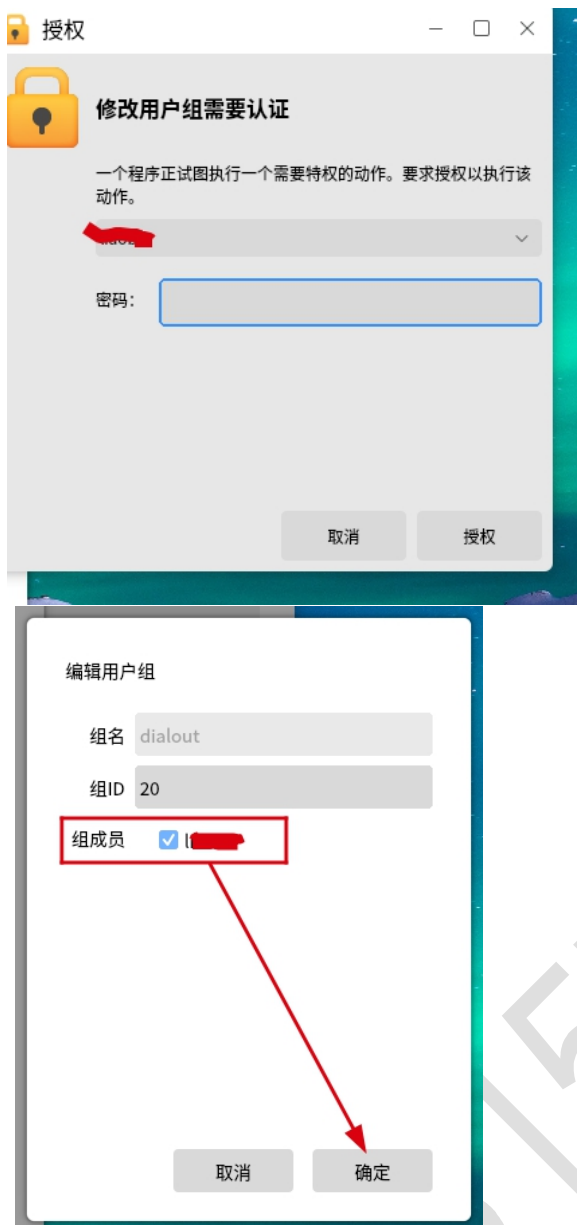
添加后：



在银河麒麟国产电脑可以使用下面的方法添加到 `dialout`

在设置--账户，当前用户-->用户组-->选择“`dialout`”-->编辑-->输入密码后-->在“组成员”勾上用户-->点“确定”，最后重启电脑。



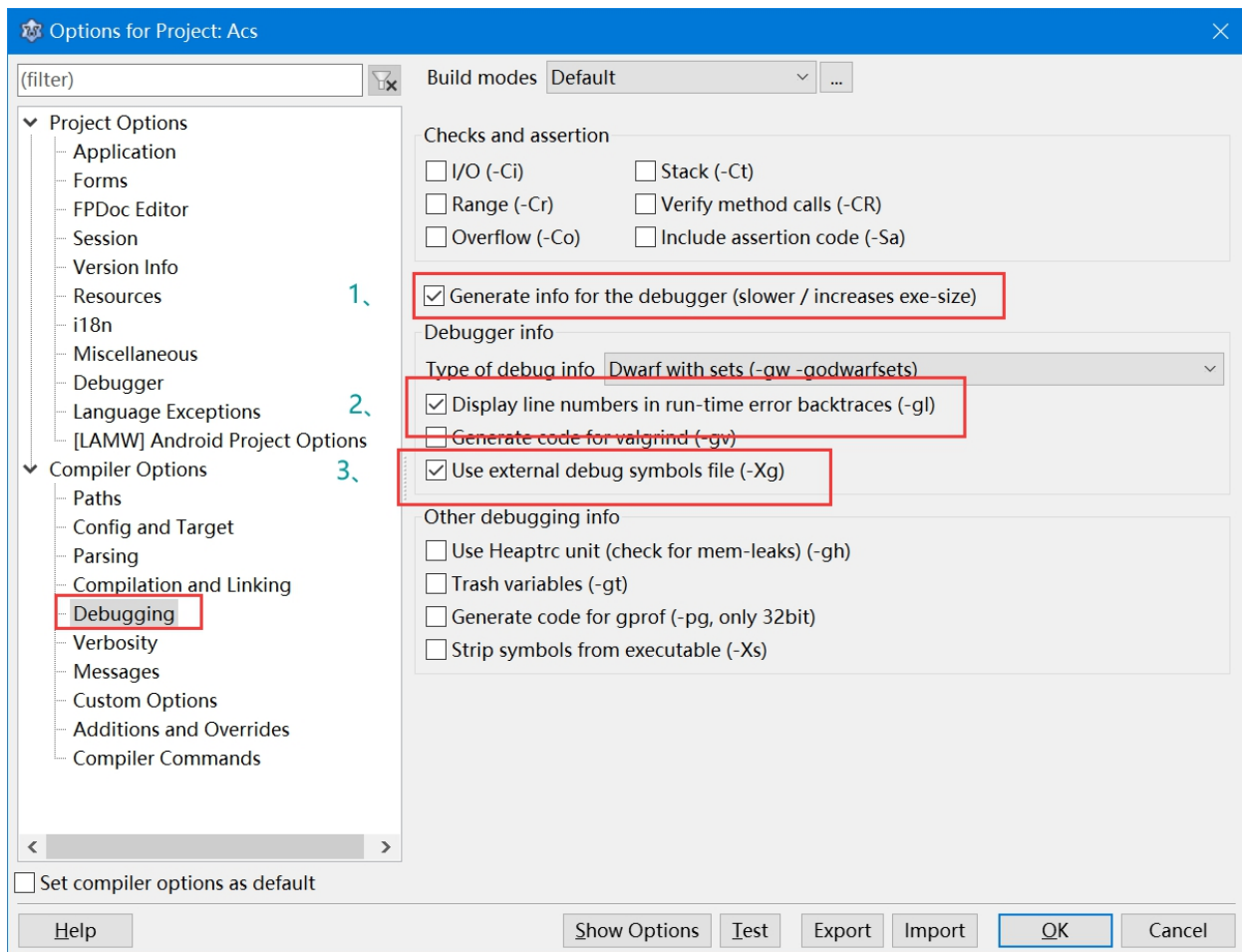


● 20、lazarus 使用技巧(2022.04.20 修正)

1、减少执行文件体积及兼顾 debug

lazarus 开启 debug 功能时，生成的执行文件非常大，如果不开启 debug 时体积比较小巧，但造成无法断点调试的情况，有没有兼顾体积和调试的方法呢？

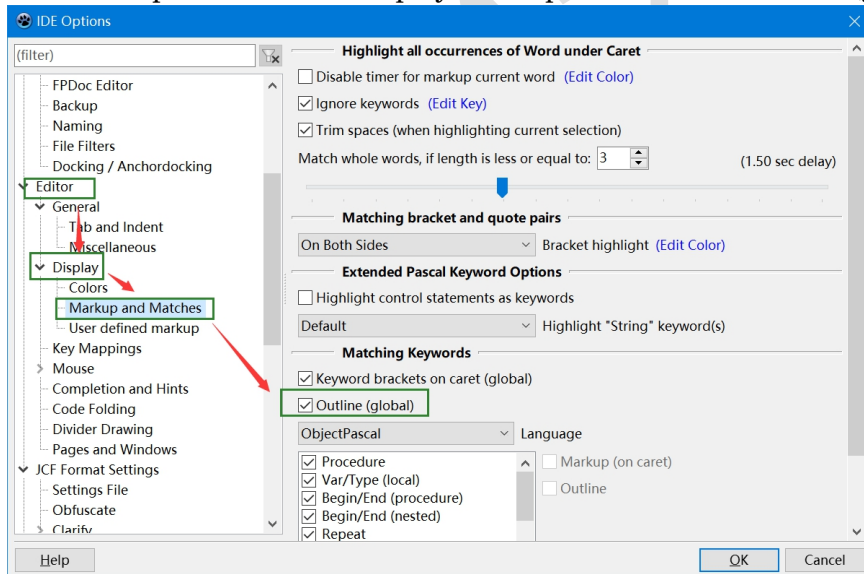
在 project--project options--compiler options--debugging--勾上 use external debug symbols file(-Xg)，重新编译应用，生成的文件体积基本和取消 debug 时差不多大小，又同时兼顾体积和实现调试。



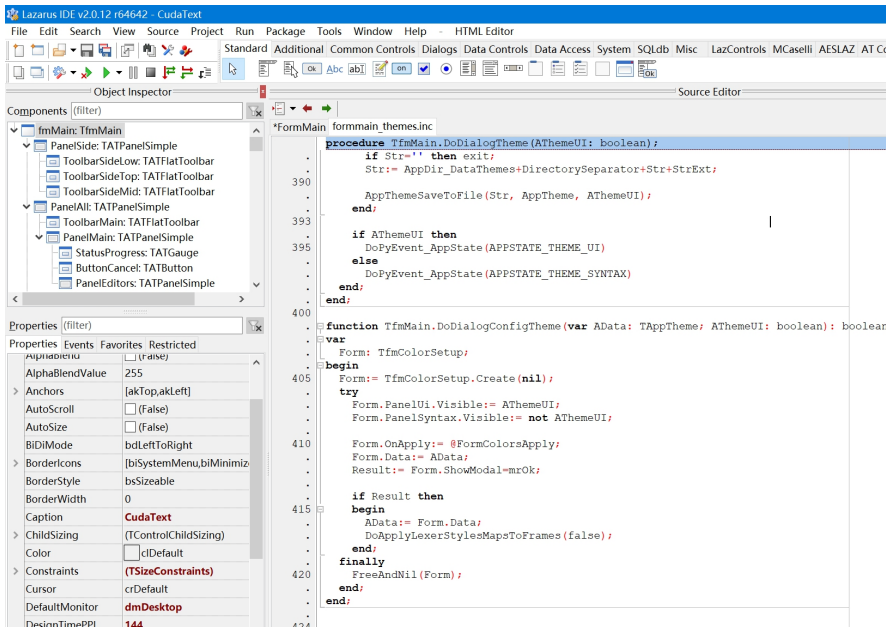
2、开启代码结构匹配连线功能

lazarus 默认没开启代码结构匹配连线功能，开启步骤：

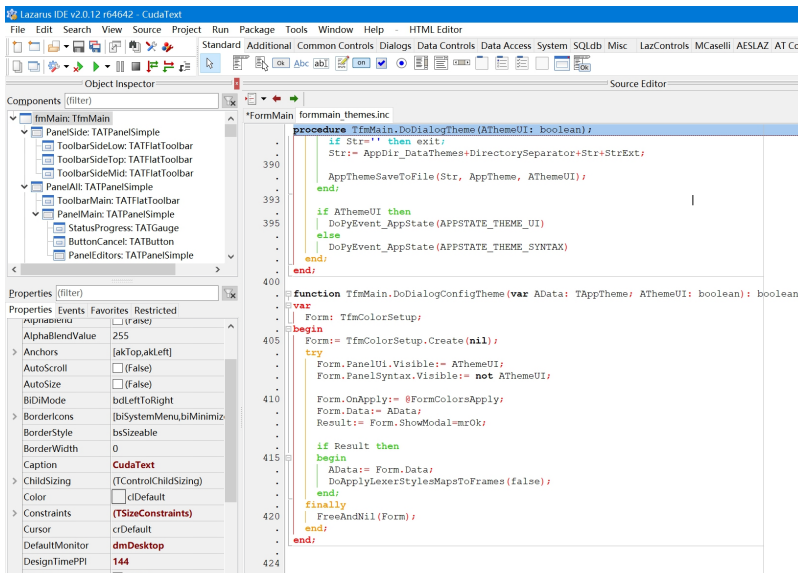
tools--Options--Editor--Display--Markup and Matches--Outline(global)



开启前：



开启后的:

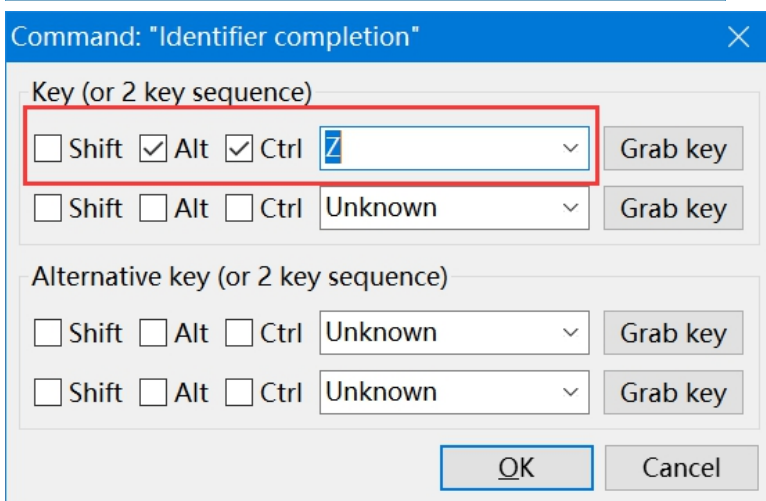
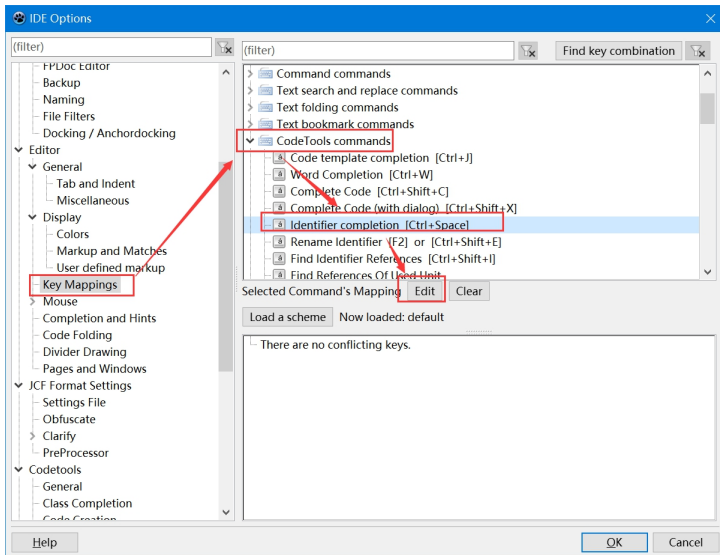


3、代码补全:

输入变量后按下 **Ctrl+W** 就可以。

4、自动完成:

默认自动完成的快捷键大多数有冲突，解决方法是打开 **Tools->Option->Editor->Key Mappings->CodeTools commands->Identifier completion**, 然后根据实际情况修改。



5、模板编辑:

按下键盘的 Ctrl+J，然后输入 class，回车。

6、多行缩进、注释:

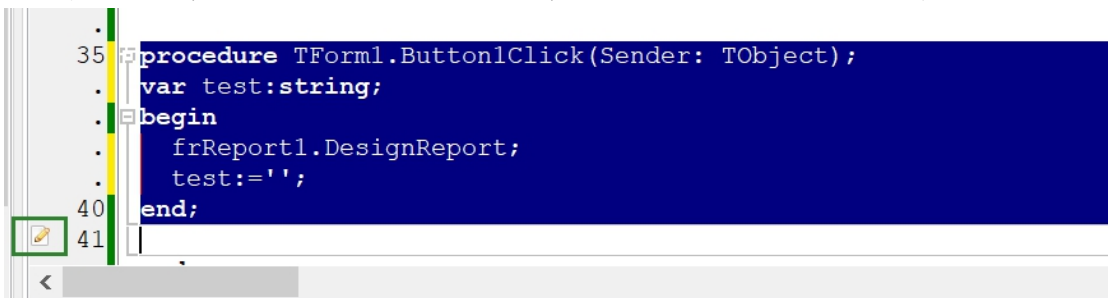
缩进的: 向左 -> Ctrl+U

向右 -> Ctrl+I (选中多行代码更爽)

多行注释: Ctrl+/*

7、选定代码后批量更改里面的单词:

先选中要修改的一段代码，等左边的笔出现后，点击编辑器左边显示行号左边的那支笔。这个时候你可以输入 test1，变量 test 就全部改为 test1，还可以按下 tab 跳到下一个单词。



```

35 procedure TForm1.Button1Click(Sender: TObject);
36 var test:string;
37 begin
38     frReport1.DesignReport;
39     test:='';
40 end;

```

将 test 修改为 test1

```

35 procedure TForm1.Button1Click(Sender: TObject);
36 var test1:string;
37 begin
38     frReport1.DesignReport;
39     test1:='';
40 end;

```

8、回车焦点转一下控件(2022.04.20 修正):

原来的方法是从网上搜到的，这个方法在界面简单时才起作用，作为通用方法应将

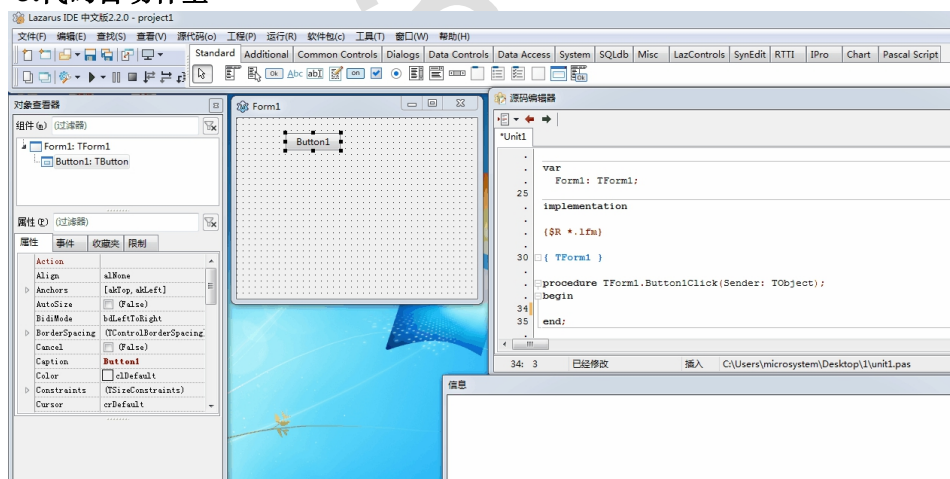
SelectNext(ActiveContorl,true,true)改为 **SelectNext(TWinControl(Sender),true,true)**，经测试发现 **TDateEdit** 日期控件不起作用。

```

procedure TForm_luru.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if (Key = VK_RETURN) or (Key = VK_Down) then
  begin
    SelectNext(TWinControl(Sender), true, true); //2022.04.20 修正
  end
  else if (Key = VK_Up) then
  begin
    SelectNext(TWinControl(Sender), false, true); //2022.04.20 修正
  end;
end;

```

9.代码自动补全



● 21、尝试解决 linux 下 Lazarus 2.2.0 版代码编辑器和 SynEdit 不支持中文输入的 Bug

linux 下 Lazarus 2.2.0 GTK2 版代码编辑器及 SynEdit 控件不支持中文输入（Lazarus for Linux QT 版支持中文输入，但发现我常用的控件不支持 QT 版，控件兼容性还是 GTK2 版比 QT 版好），长期困扰着使用中文的开发者，需然网上已有解决方案，但并不是最佳的解决办法，跟踪源码后发现，只需将这 5 个文件按下面的方法添加{\$define WITH_GTK2_IM}和{\$define Gtk2IME}就能完美修复不支持中文输入的 bug，修改后需重新编译 lazarus。

重新编译后的 lazarus 和 SynEdit 控件在 linux 已能正常输入中文，这是官方隐藏的方法，说明官方在处理 linux 等非 windows 系统不能输入中文的问题，可能还没完善，从 CudaText 的代码分析，估计 Lazarus 2.3 会支持中文输入。

存在的问题：

在 Linux(银河麒麟)使用如搜狗输入法等输入词组时只有第一个字能输入，后面的全被丢了，建议安装“fcitx-table-wbpy”输入法，这个输入法在银河麒麟和树莓派 OS 64 测试多字词组输入正常。

“fcitx-table-wbpy”输入法安装方法(在终端用执行)：

sudo apt-get install fcitx-table-wbpy

适用 lazarus 版本：lazarus 2.2.0 FPC 3.2.2

适用平台：x86 和 ARM 64 linux 【银河麒麟(CPU 为麒麟 990)和树莓派 OS 64】

不适用平台：其他 CPU 的 Linux 未测试

修改方法很简单，这 4 个文件，添加：

```
{IFDEF linux}
{$define WITH_GTK2_IM}
{$endif}
```

lazarus/components/synedit/lazsyngtk2imm.pas

lazarus/lcl/interfaces/gtk2/gtk2globals.pp

lazarus/lcl/interfaces/gtk2/gtk2widgetset.inc

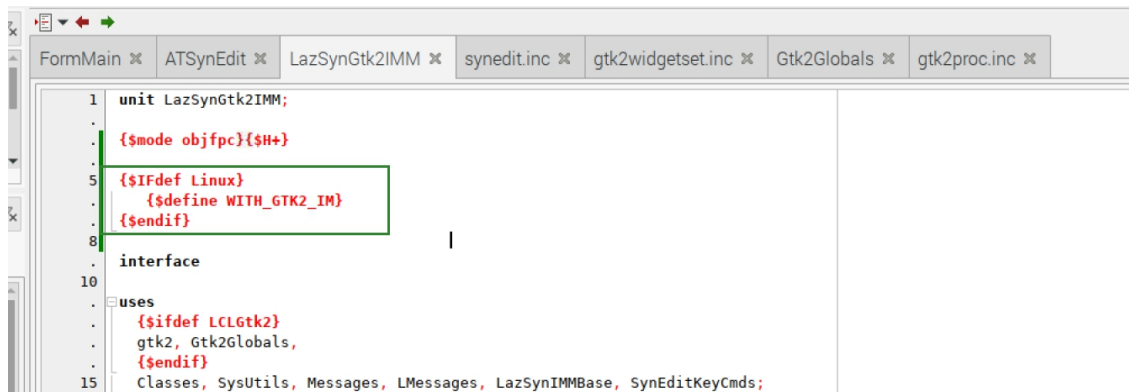
lazarus/lcl/interfaces/gtk2/gtk2proc.inc

再修改：

lazarus/components/synedit/synedit.inc 添加

```
{IFDEF linux}
{$define Gtk2IME}
{$endif}
```

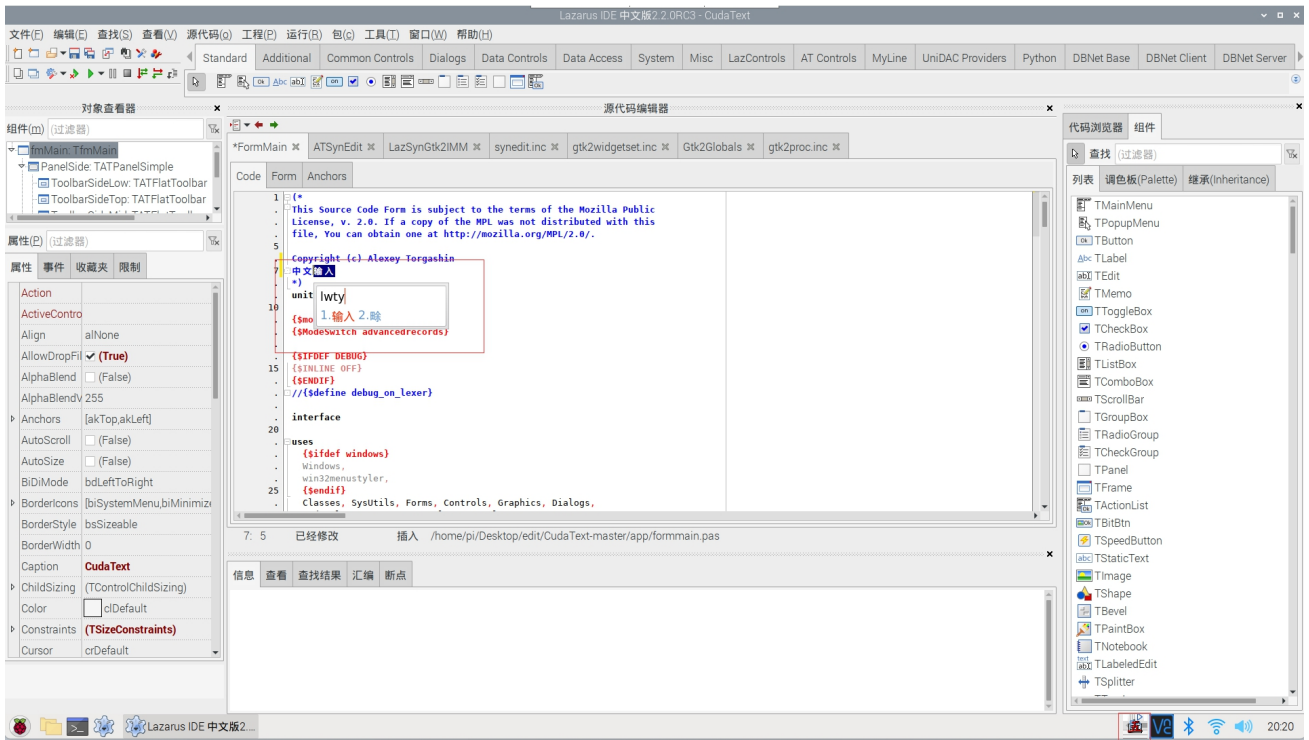
修改完成后重新编译 lazarus。



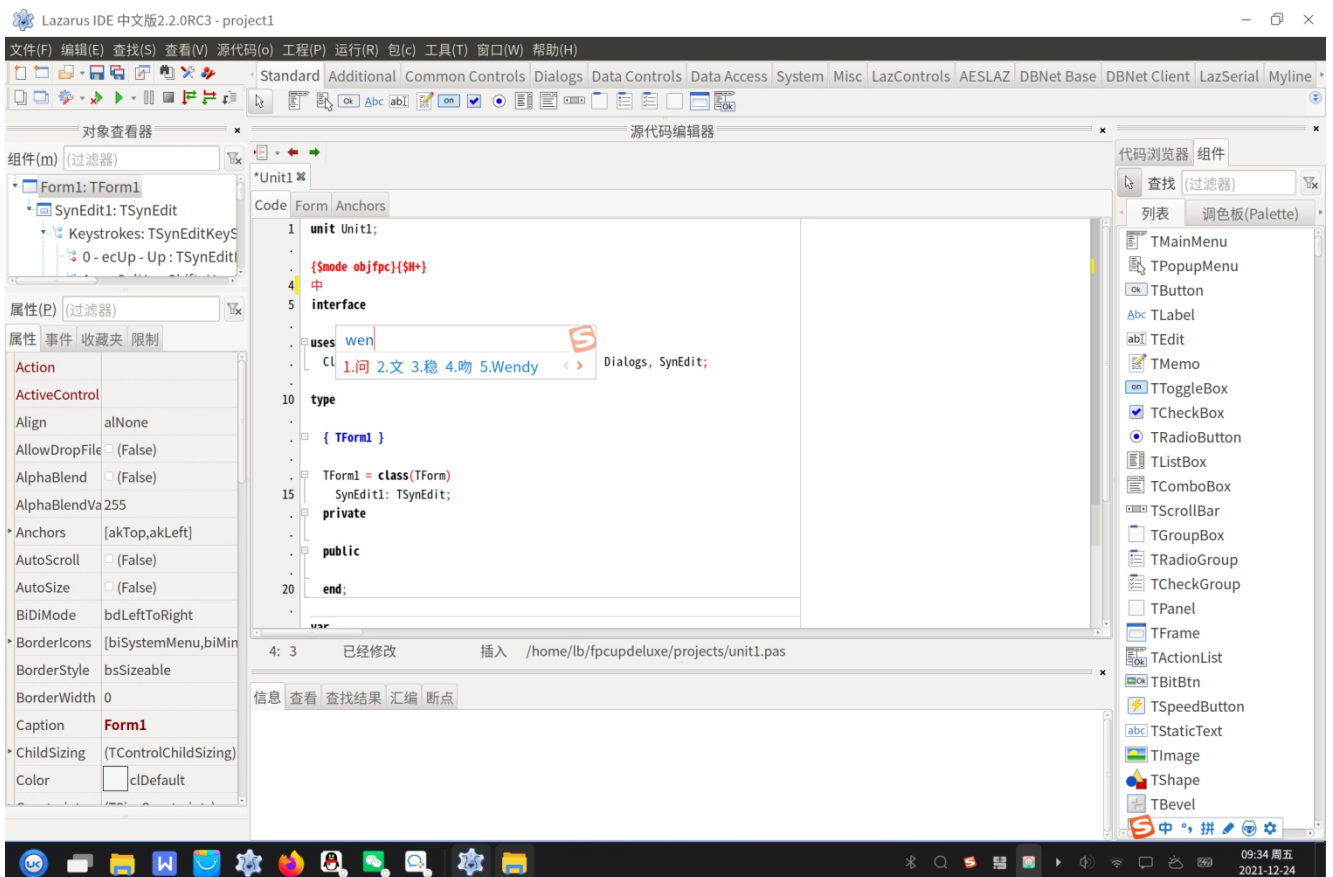

```
FormMain x ATSynEdit x LazSynGtk2IMM x synedit.inc x gtk2widgetset.inc x Gtk2Globals x gtk2proc.inc x
1 (* *****)
2 (* SynEdit Include File. This file was adapted from Brad Stowers' DFS.INC *)
3 (* file and used with permission. This will help reduce headaches when new *)
4 (* versions of Delphi and C++Builder are released, among other things. *)
5 (* *****)
6 (* Brad Stowers: bstowers@pobox.com *)
7 (* Delphi Free Stuff: http://delphifreestuff.com/ *)
8 (* February 24, 1999 *)
9 (* *****)
10 (* *)
11 (* Complete Boolean Evaluation compiler directive is turned off by including *)
12 (* this file. *)
13 (* *)
14 (* Here is a brief explanation of what each of the defines mean: *)
15 (* SYN_WIN32 : Compilation target is 32-bit Windows *)
16 (* *****)
17
18 {IFDEF FPC}
19 { $MODE OBJFPC }
20 {ENDIF}
21
22 {IFDEF Linux}
23 { $DEFINE Gtk2IME }
24 {ENDIF}
25
26 { $DEFINE SYNEDIT_INCLUDE }
```

```
FormMain x ATSynEdit x LazSynGtk2IMM x synedit.inc x gtk2widgetset.inc x Gtk2Globals x gtk2proc.inc x
1 { $MainUnit gtk2int.pas }
2 { **** }
3 { TGtk2WidgetSet }
4 { **** }
5
6 This file is part of the Lazarus Component Library (LCL)
7
8 See the file COPYING.modifiedLGPL.txt, included in this distribution,
9 for details about the license.
10 { **** }
11
12 { $IFOPT C- }
13 // Uncomment for local trace
14 // { $C+ }
15 // { $DEFINE ASSERT_IS_ON }
16 { $ENDIF }
17
18 { $IFDEF Linux }
19 { $define WITH_GTK2_IM }
20 { $endif }
21
22 { $IFDEF USE_GTK_MAIN_OLD_ITERATION }
23 var
24 Gtk2MPF: TGPollFunc;
```

```
FormMain x ATSynEdit x LazSynGtk2IMM x synedit.inc x gtk2widgetset.inc x Gtk2Globals x gtk2proc.inc x
1 {
2 ****
3 This file is part of the Lazarus Component Library (LCL)
4
5 See the file COPYING.modifiedLGPL.txt, included in this distribution,
6 for details about the license.
7 ****
8 }
9
10 unit Gtk2Globals;
11
12 { $mode objfpc } { $H+ }
13
14 { $IFDEF Linux }
15 { $define WITH_GTK2_IM }
16 { $endif }
17
18 interface
```

银河麒麟中文输入：



● 22、UNIDAC 在 arm linux 运行出错解决方法

在网友“努力的干”指点下解决了 unidac 不能在 arm linux 运行出错的问题。解决方法也很简单，只需修改 project source，在 uses 加上

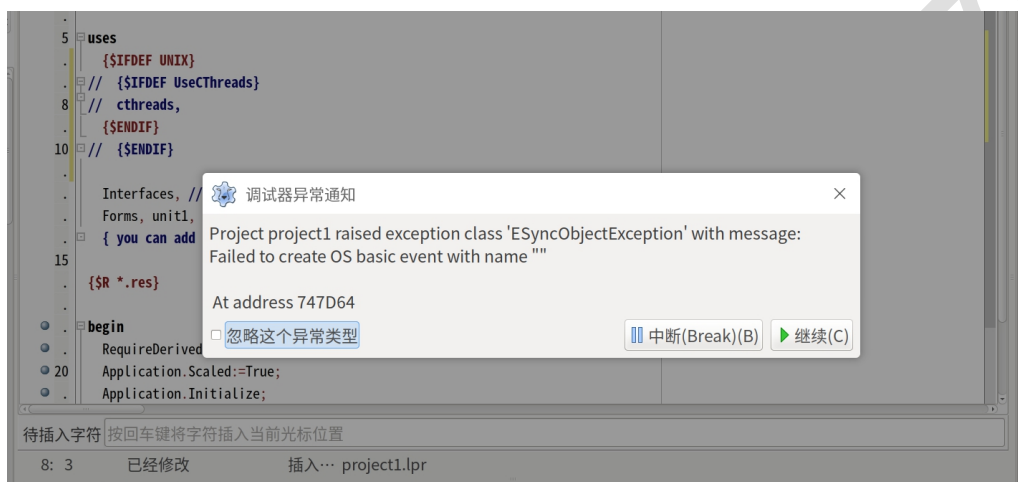
```

{$IFDEF UNIX}
  cthreads,
{$ENDIF}

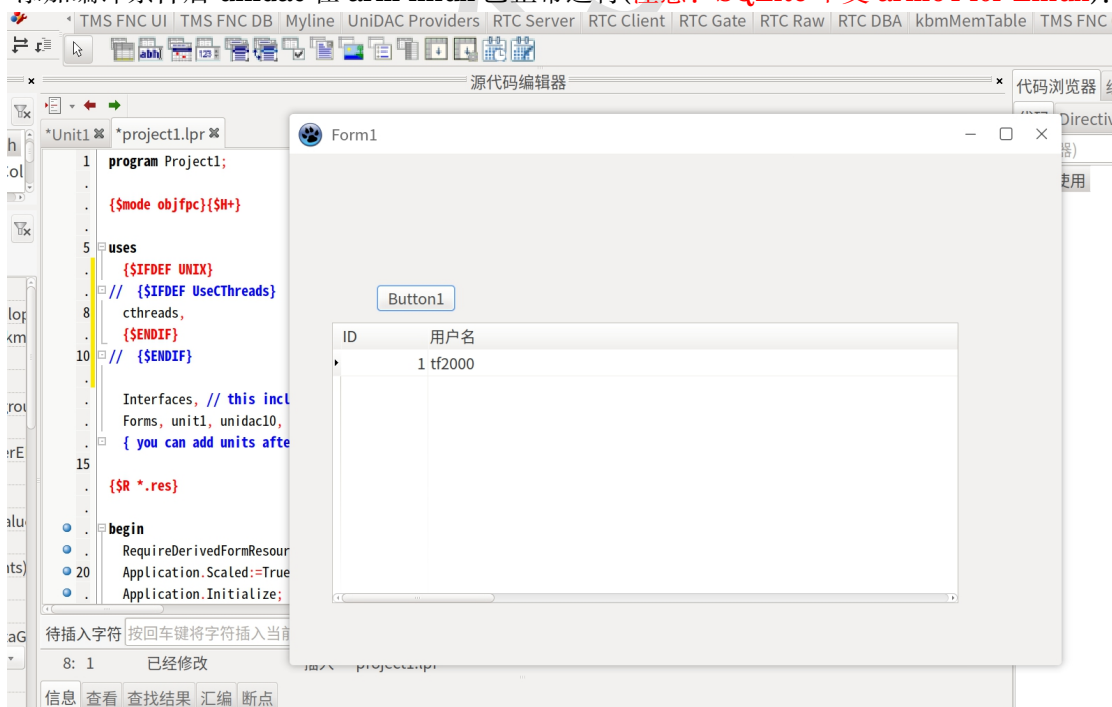
```



```
*Unit1 *project1.lpr
1 program Project1;
.
. {$mode objfpc}{$H+}
.
5 uses
. {$IFDEF UNIX}
7 // {$IFDEF UseCThreads}
. cthreads,
. {$ENDIF}
10 // {$ENDIF}
.
. Interfaces, // this includes the LCL widgetset
. Forms, unit1, unidac10, pgprovider10
. { you can add units after this };
15
. {$R *.res}
.
begin
```



添加编译条件后 unidac 在 arm linux 已正常运行(注意: SQLite 不支 arm64 for Linux):



● 23、lazarus 交叉编译步骤

1.推荐用 fpcupdeluxe 安装 lazarus, 从 <https://github.com/newpascal/fpcupdeluxe/releases/latest> 下载适合的版本

如果第一次在 linux 运行 fpcupdeluxe 需配置一下运行环境:

GNU make

the binutils (make etc); e.g. in a package called build-essential

GIT client

bunzip2 (probably present in most distributions)

unzip

untar

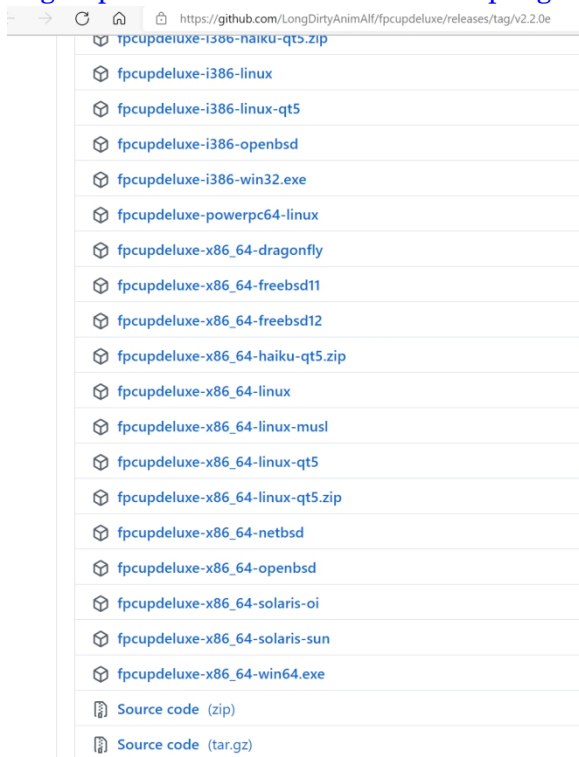
subversion client: svn

gdb is not needed for building FPC/Lazarus but needed for debugging your Lazarus programs

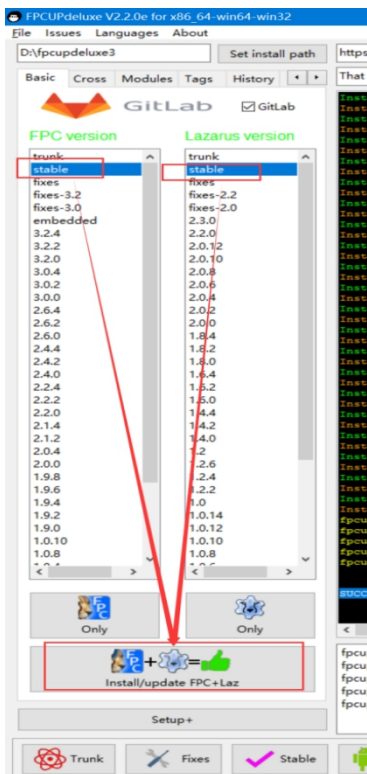
libX11, libgdk_pixbuf-2.0, libpango-1.0, libgdk-x11-2.0

E.g. on Debian or Ubuntu, do something like:

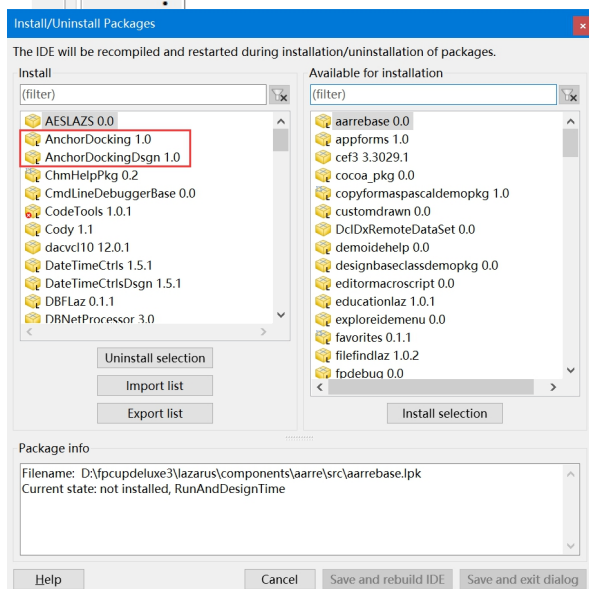
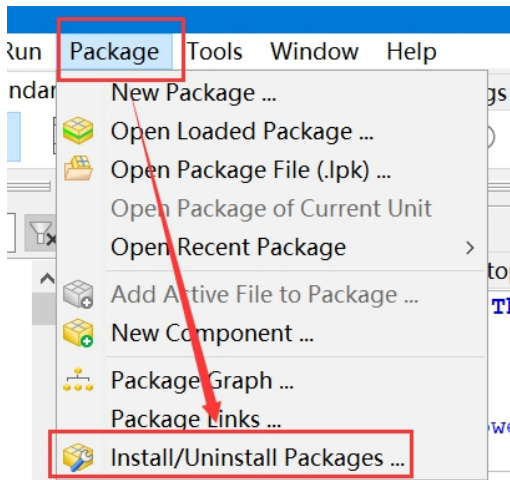
`sudo apt install make binutils git build-essential gdb subversion zip unzip libx11-dev libgtk2.0-dev libgdk-pixbuf2.0-dev libcairo2-dev libpango1.0-dev git`



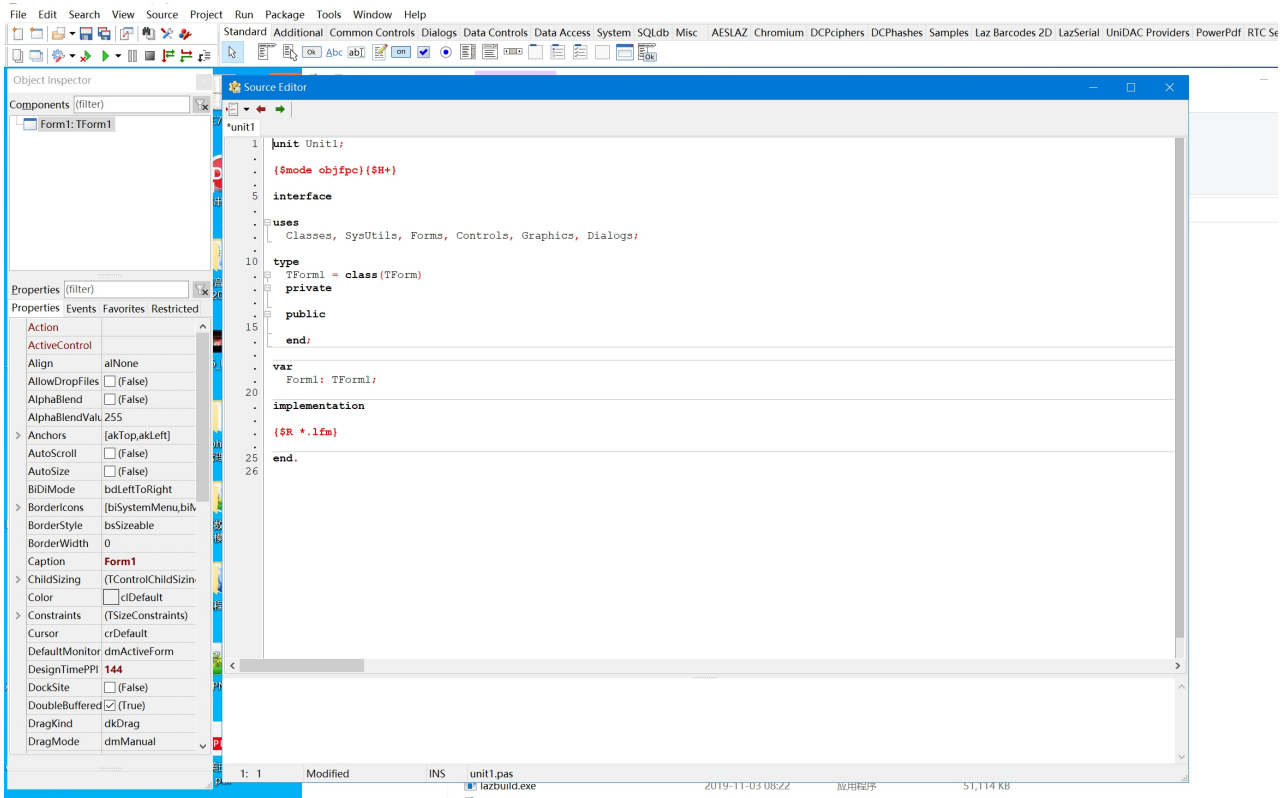
建议 FPC 和 Lazarus 选择 stable



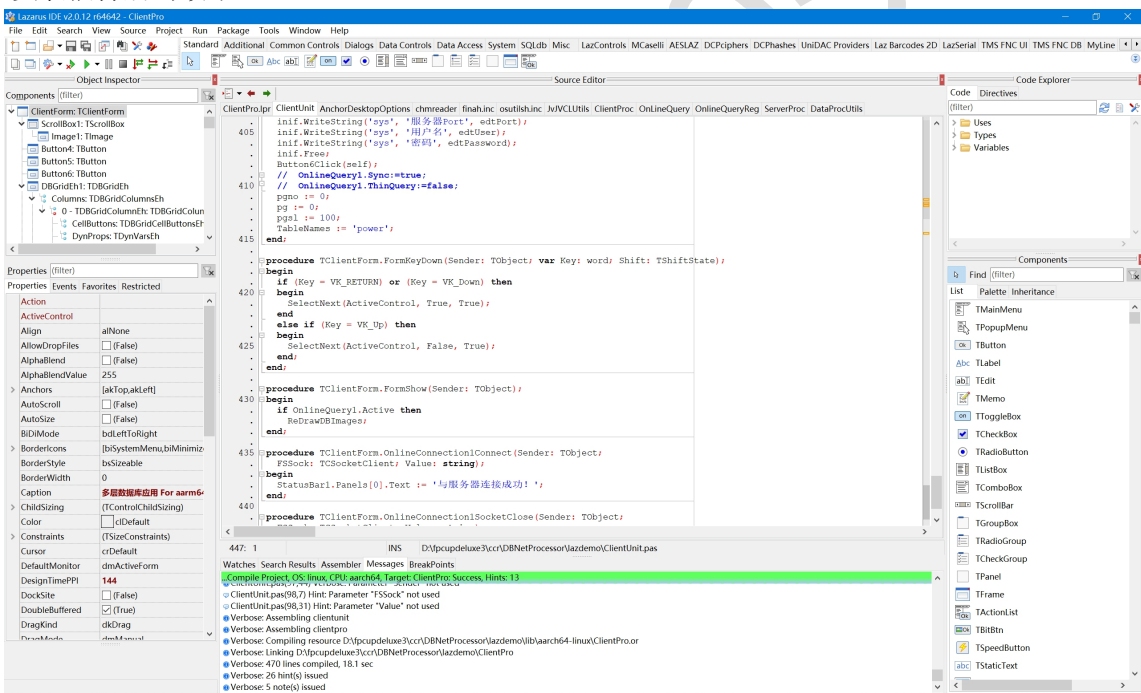
安装成功后可以安装下面 2 个插件，Lazarus 的界面就像 delphi XE 一样



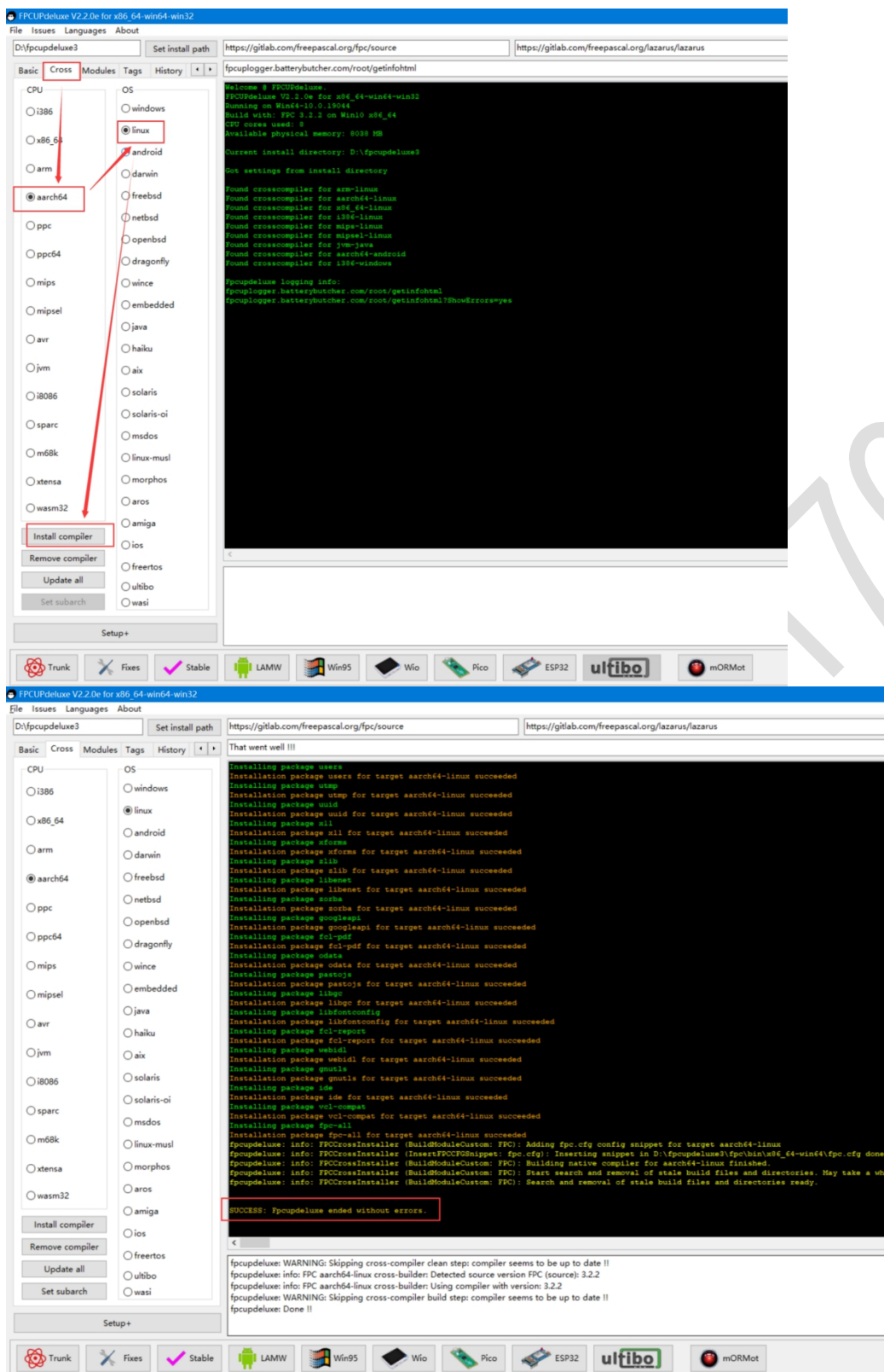
安装插件前的界面和 D7 一样：



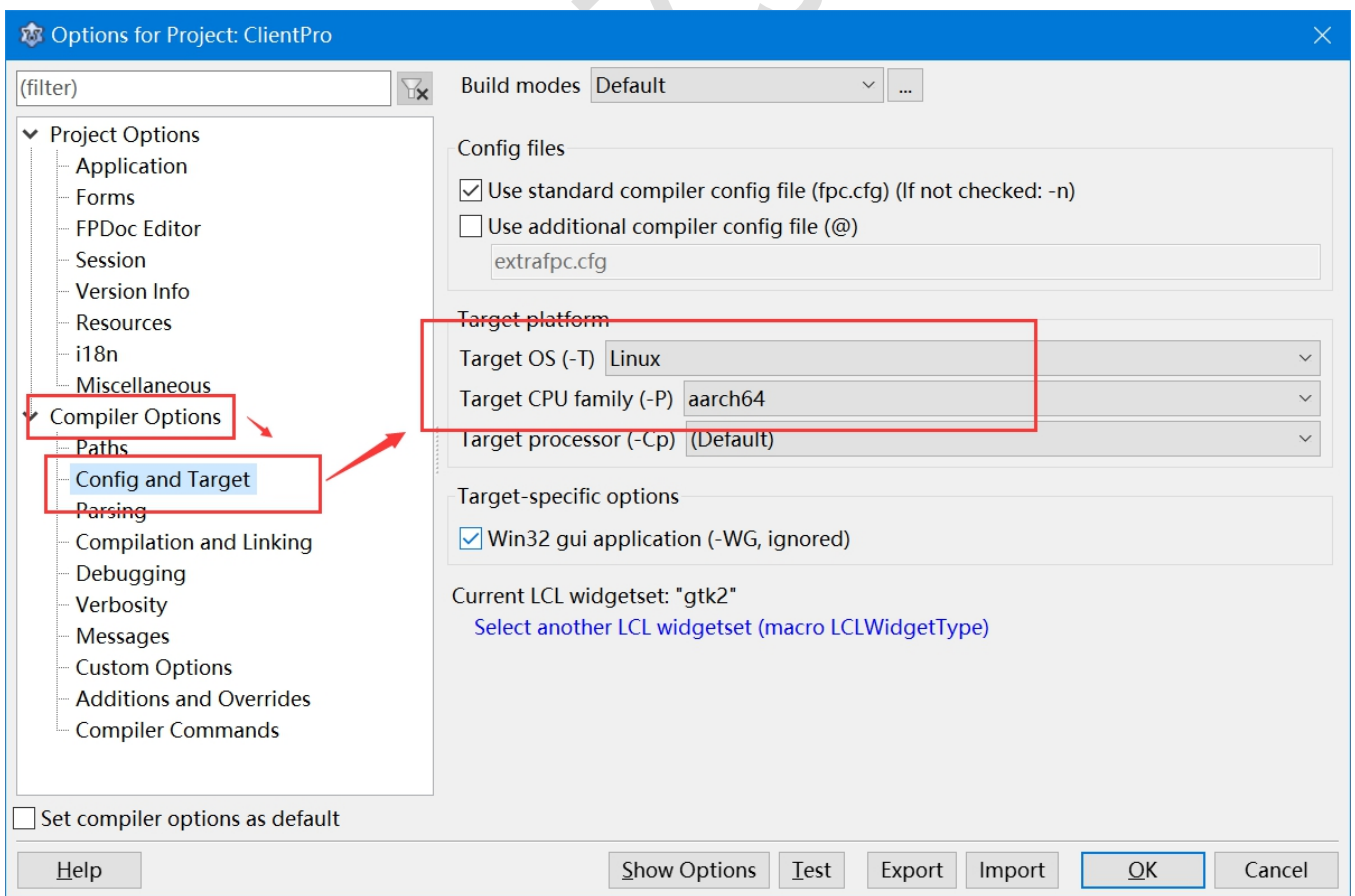
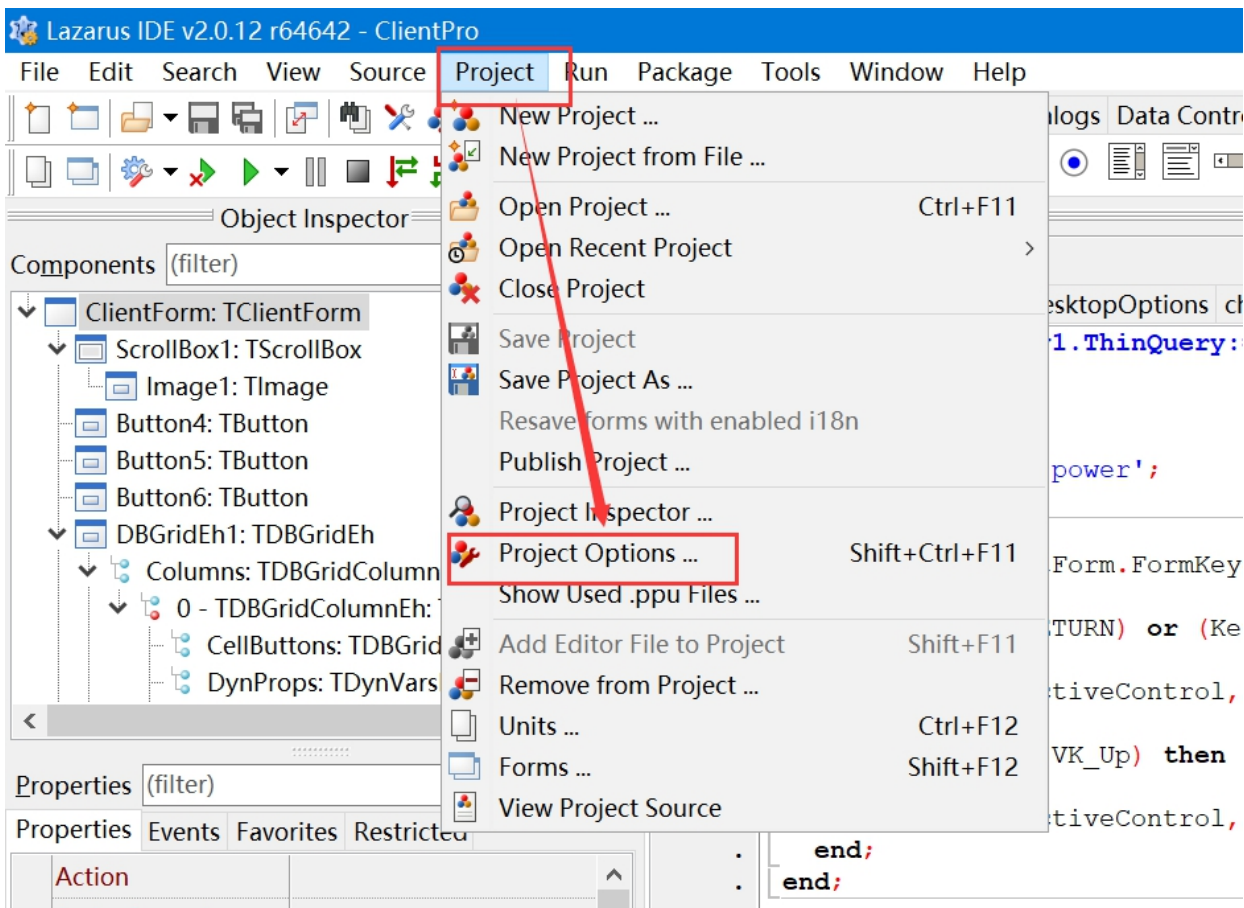
安装插件后的效果：



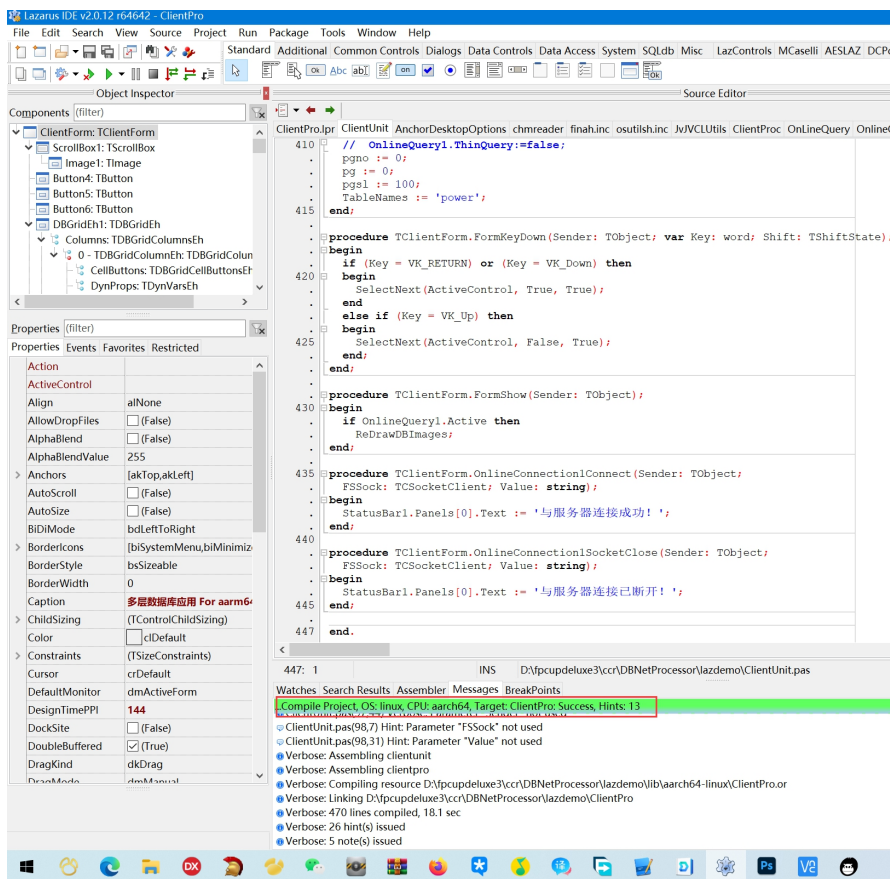
2.lazarus 安装成功后，切换到 Cross 页面，选择需要的 CPU 和 OS，然后点"install compiler",最后看到: SUCCESS: Fpcupdeluxe ended without errors. 指定平台就添加成功。



3、打开 lazarus--Project--Project Option--Compiler Options--Config and Target--Target platform 选择生成的 OS 和 CPU



4、最后编译就能生成指定平台的执行文件。



在 windows 编译后程序在树梅派运行的效果：

回弹法混凝土强度检测

原始记录 报告预览

报告编号	2021-12		
上一编号	检测日期	见证类型	
委托单位	检测依据		
工程名称	标称动能		
备注	芯样修正系数	<input type="checkbox"/> 泵送混凝土	
结构或构件名称			
强度等级	C	灌注日期	角度
测区	测点数据		碳化深度 (mm)

报告编号	构件名称	序号	强度等级	角度	测面	回弹值1	回弹值2	回弹值3	回弹值4

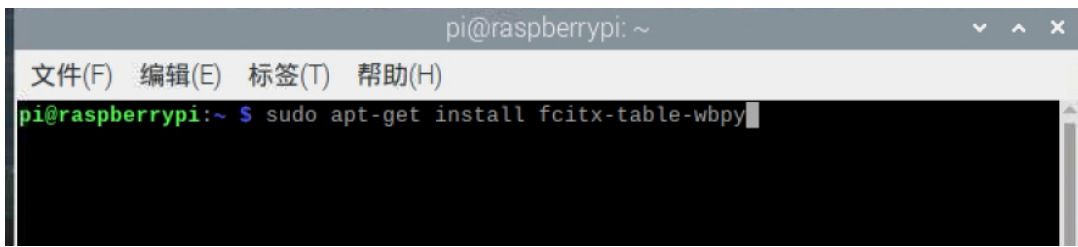
下一编号 资料查找 ☐ 重新计算 批量打印 打印 预览 修改报告 关闭 保存

回弹法混凝土(砂浆)... 回弹法混凝土强度检测

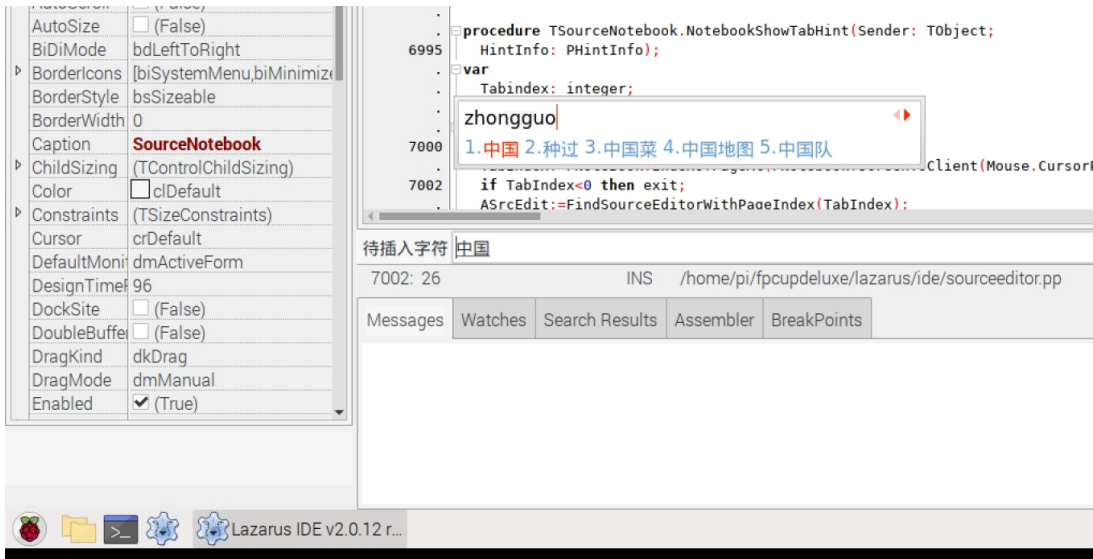
● 24、raspberry 添加拼音五笔输入法

在终端安装 fcit-table-wbpy

sudo apt-get install fcitx-table-wbpy



安装后运行效果，这个输入法很不错，可以直接输入拼音五笔，不用切换。



● 25、银河麒麟(aarch64)安装 Lazarus 步骤

从 <https://github.com/newpascal/fpcupdeluxe/releases/latest> 下载最新的 fpcupdeluxe-aarch64-linux

1、在终端安装 Lazarus 运行环境：

```
sudo apt-get install gdb
```

```
sudo apt-get install svn
```

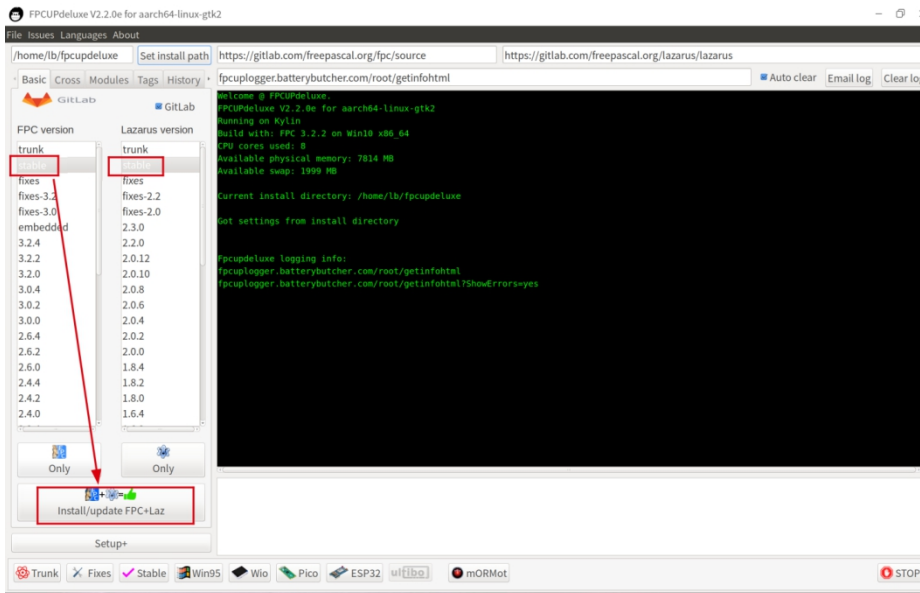
```
sudo apt-get install git
```

```
sudo apt-get install libgtk2.0-dev
```

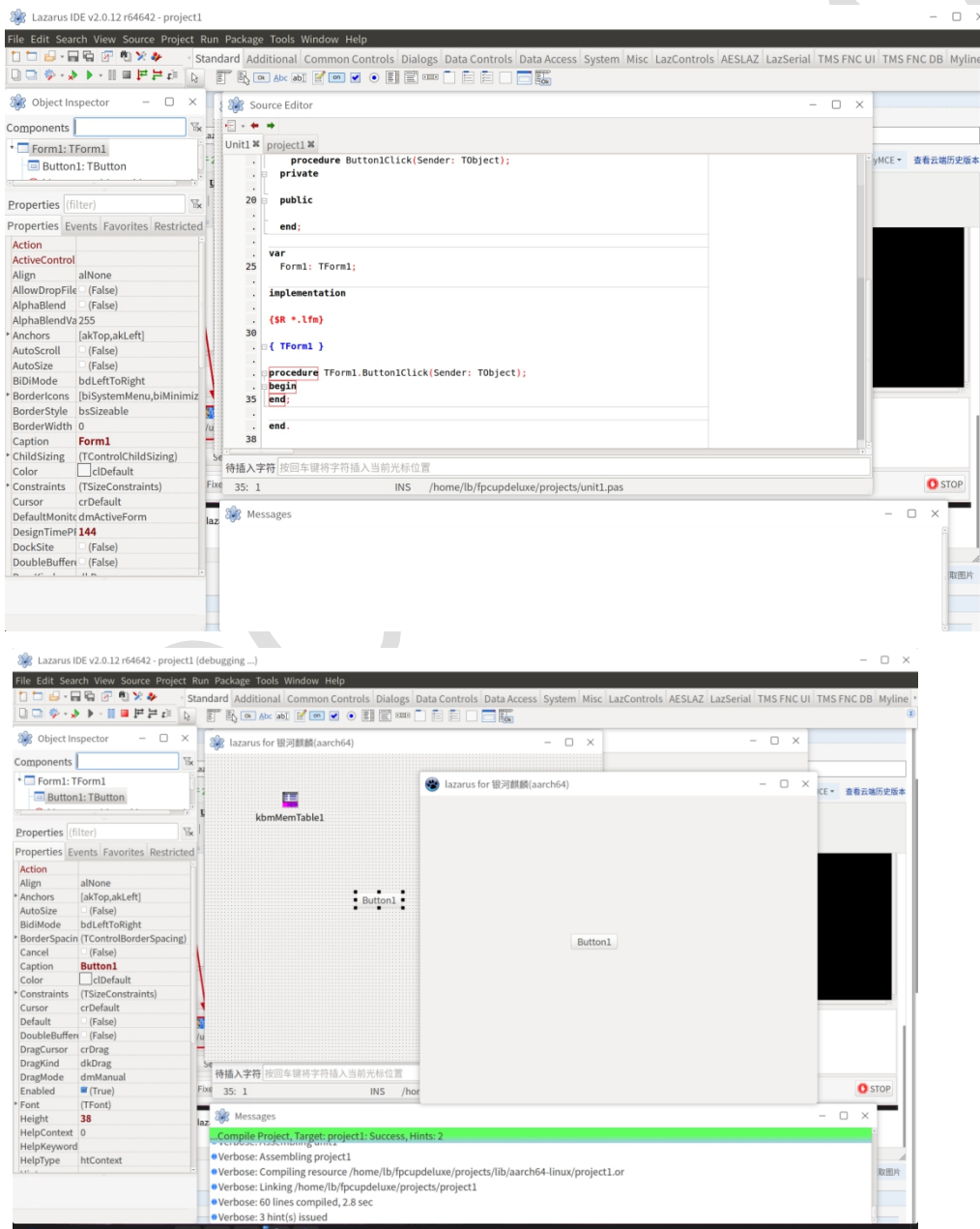
```
sudo apt-get install libgdk-pixbuf2.0-dev
```

2、运行 fpcupdeluxe-aarch64-linux 安装 lazarus

FPC 和 Lazarus 的版本选择 stable，然后点击 install



3、安装成功后的 lazarus:



● 26、Lazarus Linux 创建菜单及桌面快捷方式

加上下面的代码后，Lazarus 编译的程序运行时会自动创建菜单及桌面快捷方式，在菜单的快捷方式可直接打开应用（在中标麒麟和树莓派测试使用正常），还解决了每次运行时都会出现下图提示的问题。

正在打开一个可执行文件ClientPro, 你希望?

使用默认打开方式

直接运行

在终端运行

取消



```
{IFDEF LINUX}
uses
  process,
  BaseUnix;
{$ENDIF LINUX}

procedure TClientForm.FormCreate(Sender: TObject);
var
  vFileName: string;
  aDirectory,Output: string;
  iniFile: TIniFile;
begin
{$IFDEF UNIX}
  aDirectory := ConcatPaths([GetUserDir, '.local', 'share', 'applications']);
  if DirectoryExists(aDirectory) then
    try
      if DirectoryExists(aDirectory) then
        begin
```

```

vFileName := aDirectory + DirectorySeparator + ExtractFileName('回弹检测.desktop');
IniFile := TIniFile.Create(vFileName);
try
    iniFile.WriteString('Desktop Entry', 'Encoding', 'UTF-8');
    iniFile.WriteString('Desktop Entry', 'Name', '回弹检测');
    iniFile.WriteString('Desktop Entry', 'Comment', '回弹检测计算软件');
    iniFile.WriteString('Desktop Entry', 'Exec', Format('%s', [ParamStr(0)]));
    iniFile.WriteString('Desktop Entry', 'Type', 'Application');
    iniFile.WriteString('Desktop Entry', 'GenericName', '回弹检测');
    iniFile.WriteString('Desktop Entry', 'Category', '工程检测;');
    iniFile.WriteString('Desktop Entry', 'Categories', '工程检测;');
    iniFile.WriteString('Desktop Entry', 'Terminal', 'false');
    iniFile.WriteString('Desktop Entry', 'Icon', ExtractFilePath(ParamStr(0)) + '回弹检.png');
finally
    IniFile.Free;
end;
FpChmod(vFileName, &755);
aDirectory:=ConcatPaths([GetUserDir,'.local','share','applications']);
RunCommand('update-desktop-database' ,[aDirectory],Output,[poUsePipes, poStderrToOutPut]){IF
DEFINED(FPC_FULLVERSION) AND (FPC_FULLVERSION >= 30200)},swoHide{$ENDIF});
//拷贝"回弹检测.desktop"到桌面，并设置为“可执行”
aDirectory:=ConcatPaths([GetUserDir,'桌面']);
if not DirectoryExists(aDirectory) then
    aDirectory:=ConcatPaths([GetUserDir,'Desktop']);
FileUtil.CopyFile(vFileName,aDirectory+DirectorySeparator+ExtractFileName(vFileName),[]);
FpChmod(aDirectory+DirectorySeparator+ExtractFileName(vFileName), &755);
end;
except
end;
{$ENDIF}
end;

```