

Mathematical Expression Project

Kompilacja – pliki makefile

1. Program skompilowany został w środowisku Code::Blocks. Napisany w całości w języku C.
2. Używana jest biblioteka zewnętrzna GTK ze szczególnym naciskiem na obiekt Cairo, który wykorzystywany jest jako główna część graficzna projektu.
3. Dołączane są biblioteki zewnętrzne:
 - a. 'gtk/gtk.h' - gdyż jest to nasza podstawa baza do tworzenia aplikacji okienkowej.
 - b. 'stdio.h' - gdyż zawiera definicję NULL.
 - c. 'stdbool.h' - zawiera typ boolowski.
4. W środowisku File Encoding powinno być ustawione na UTF-8.
5. Opcje kompilacji w kompilatorze gcc powinny być ustawione na:
"gcc -std=c99 -Wall 'pkg-config gtk+-3.0 --cflags --libs'"
6. Można też używać makefile'a, przykład wygenerowany przez cbp2make.exe:

```
#-----#
# This makefile was generated by 'cbp2make' tool rev.147 #
#-----#
WORKDIR = %cd%

CC = gcc.exe
CXX = g++.exe
AR = ar.exe
LD = g++.exe
WINDRES = windres.exe

INC =
CFLAGS = -Wall
RESINC =
LIBDIR =
LIB =
LDFLAGS =

INC_DEBUG = $(INC)
CFLAGS_DEBUG = $(CFLAGS) -g
RESINC_DEBUG = $(RESINC)
RCFLAGS_DEBUG = $(RCFLAGS)
LIBDIR_DEBUG = $(LIBDIR)
LIB_DEBUG = $(LIB)
LDFLAGS_DEBUG = $(LDFLAGS)
OBJDIR_DEBUG = obj\\Debug
DEP_DEBUG =
OUT_DEBUG = bin\\Debug\\TEX\ Project.exe

INC_RELEASE = $(INC)
CFLAGS_RELEASE = $(CFLAGS) -O2
RESINC_RELEASE = $(RESINC)
RCFLAGS_RELEASE = $(RCFLAGS)
```

```

LIBDIR_RELEASE = $(LIBDIR)
LIB_RELEASE = $(LIB)
LDFLAGS_RELEASE = $(LDFLAGS) -s
OBJDIR_RELEASE = obj\\Release
DEP_RELEASE =
OUT_RELEASE = bin\\Release\\TEX\ Project.exe

OBJ_DEBUG = $(OBJDIR_DEBUG)\\draw.o $(OBJDIR_DEBUG)\\main.o $(OBJDIR_DEBUG)\\stack.o
$(OBJDIR_DEBUG)\\tex.o $(OBJDIR_DEBUG)\\tree.o

OBJ_RELEASE = $(OBJDIR_RELEASE)\\draw.o $(OBJDIR_RELEASE)\\main.o
$(OBJDIR_RELEASE)\\stack.o $(OBJDIR_RELEASE)\\tex.o $(OBJDIR_RELEASE)\\tree.o

all: debug release

clean: clean_debug clean_release

before_debug:
cmd /c if not exist bin\\Debug md bin\\Debug
cmd /c if not exist $(OBJDIR_DEBUG) md $(OBJDIR_DEBUG)

after_debug:

debug: before_debug out_debug after_debug

out_debug: before_debug $(OBJ_DEBUG) $(DEP_DEBUG)
$(LD) $(LIBDIR_DEBUG) -o $(OUT_DEBUG) $(OBJ_DEBUG) $(LDFLAGS_DEBUG) $(LIB_DEBUG)
$(OBJDIR_DEBUG)\\draw.o: draw.c
$(CC) $(CFLAGS_DEBUG) $(INC_DEBUG) -c draw.c -o $(OBJDIR_DEBUG)\\draw.o
$(OBJDIR_DEBUG)\\main.o: main.c
$(CC) $(CFLAGS_DEBUG) $(INC_DEBUG) -c main.c -o $(OBJDIR_DEBUG)\\main.o
$(OBJDIR_DEBUG)\\stack.o: stack.c
$(CC) $(CFLAGS_DEBUG) $(INC_DEBUG) -c stack.c -o $(OBJDIR_DEBUG)\\stack.o
$(OBJDIR_DEBUG)\\tex.o: tex.c
$(CC) $(CFLAGS_DEBUG) $(INC_DEBUG) -c tex.c -o $(OBJDIR_DEBUG)\\tex.o
$(OBJDIR_DEBUG)\\tree.o: tree.c
$(CC) $(CFLAGS_DEBUG) $(INC_DEBUG) -c tree.c -o $(OBJDIR_DEBUG)\\tree.o

clean_debug:
cmd /c del /f $(OBJ_DEBUG) $(OUT_DEBUG)
cmd /c rd bin\\Debug
cmd /c rd $(OBJDIR_DEBUG)

before_release:
cmd /c if not exist bin\\Release md bin\\Release
cmd /c if not exist $(OBJDIR_RELEASE) md $(OBJDIR_RELEASE)

after_release:

release: before_release out_release after_release

```

```
out_release: before_release $(OBJ_RELEASE) $(DEP_RELEASE)
$(LD) $(LIBDIR_RELEASE) -o $(OUT_RELEASE) $(OBJ_RELEASE) $(LDFLAGS_RELEASE) $(LIB_RELEASE)

$(OBJDIR_RELEASE)\\draw.o: draw.c
$(CC) $(CFLAGS_RELEASE) $(INC_RELEASE) -c draw.c -o $(OBJDIR_RELEASE)\\draw.o

$(OBJDIR_RELEASE)\\main.o: main.c
$(CC) $(CFLAGS_RELEASE) $(INC_RELEASE) -c main.c -o $(OBJDIR_RELEASE)\\main.o

$(OBJDIR_RELEASE)\\stack.o: stack.c
$(CC) $(CFLAGS_RELEASE) $(INC_RELEASE) -c stack.c -o $(OBJDIR_RELEASE)\\stack.o

$(OBJDIR_RELEASE)\\tex.o: tex.c
$(CC) $(CFLAGS_RELEASE) $(INC_RELEASE) -c tex.c -o $(OBJDIR_RELEASE)\\tex.o

$(OBJDIR_RELEASE)\\tree.o: tree.c
$(CC) $(CFLAGS_RELEASE) $(INC_RELEASE) -c tree.c -o $(OBJDIR_RELEASE)\\tree.o

clean_release:
cmd /c del /f $(OBJ_RELEASE) $(OUT_RELEASE)
cmd /c rd bin\\Release
cmd /c rd $(OBJDIR_RELEASE)

.PHONY: before_debug after_debug clean_debug before_release after_release clean_release
```

Jakub Kuczkowiak - 11.02.2017