

Open Source:
Some Assembly Required

Greg Smith

gsmith@well.com

Presented to the
Southern California OS/2 Users Group

January 18, 2003

Outline of Presentation

HPCalc as an example of open source

Working with large projects

The development environment

Installing the software

Building the application

Demonstration

HPCalc

Program Files

hpcalc.c, hpcalc.h – main program

change.c, change.h – redraw routines

hex-oct.c, hex-oct.h – conversions

ini.c, ini.h – initialization

interface.c, interface.h – the guts of it all

buttons.h – button definitions

interface-vars.h – definitions for interface routines

HPCalc (Continued)

Resource Files

hpclcd.dlg – dialog definitions

hpclc.rc -- resources

Make file

Approximately 4000 lines of code (65 to 70 pages)

Code Snippet – hpcalc.c

```
/* HP Calculator */
/* Definitions */
#define INCL_PM
#define INCL_DOS
#define INCL_WINHELP

#define MYAPPCLASS          "HPcalc"

/* Includes */
#include <stdio.h>
#include <string.h>
#include <io.h> /*used by emx instead of unistd.h */
/* #include <unistd.h> */
#include <os2.h>
#include "HPcalc.h"
#include "interface.h"
#include "ini.h"
#include "change.h"

/* Local Prototypes */
MRESULT EXPENTRY ClientWinProc (HWND, ULONG, MPARAM, MPARAM);
MRESULT EXPENTRY SubclassFrame (HWND, ULONG, MPARAM, MPARAM);
MRESULT EXPENTRY HexOctWinProc (HWND, ULONG, MPARAM, MPARAM);
void GetIniFilename(int argc, char *argv[], PSZ buf);
void GetFullPathToExe(PSZ buf);
BOOL ConvertExeToIni(PSZ buf);

/* Local Global Variables */
PFNWP    pfnFrame;
HINI     hini;
PSZ      szIniFile="HPcalc.ini";
```

Code Snippet – hpcalc.c

```
/* main function */

void main (int argc, char *argv[], char *envp[])
{
    HWND          hwndMenu;
    HMQ           hmq;
    QMSG          qmsg;
    SWP           swp, swpm;
    ULONG         flCreate;
    FRAMECDATA    fcdata;
    USHORT        usWinSize=SIZE_NORM;
    BOOL          fIniRead=FALSE, fXyztMode=FALSE;
    CHAR          buf[256];
    APIRET        rc;

    SetPositions(SIZE_NORM);

    if ( !(hab = WinInitialize (0)) )
        return;

    GetIniFilename(argc, argv, buf);

    if ((hini = PrfOpenProfile(hab, buf)) ) {
        ReadIniFile(hini, &fIniRead, &usWinSize, &fXyztMode, &swp);
    }

    hmq = WinCreateMsgQueue (hab, 0);
    WinRegisterClass(
        hab,
        MYAPPCLASS,
        (PFNWP) ClientWinProc,
        CS_SIZEREDRAW,
        0);
}
```

Code Snippet – hpcalc.rc

```
/* HPcalc.rc */

#include <os2.h>
#include "HPcalc.h"

ICON ID_HPCALC HPcalc.ico
/* icon number does not matter, but it has to be the first icon */

MENU ID_HPCALC
BEGIN
    SUBMENU "~File",                IDM_FILE
    BEGIN
        MENUITEM "E~xit\tF3",        IDM_EXIT
        MENUITEM SEPARATOR
        SUBMENU "~Configuration",    IDM_CONF
        BEGIN
            MENUITEM "~XYZT Mode",    IDM_XYZT
            MENUITEM "Small ~Window", IDM_SIZE
            MENUITEM "1000s ~Separators",IDM_COMMA
        /*
            MENUITEM "~Background Color",    IDM_BACKCOLOR, MIA_DISABLED
            MENUITEM "Shift ~Text Color",    IDM_SHIFTCOLOR, MIA_DISABLED
        */
        END
        MENUITEM SEPARATOR
        MENUITEM "~About HPcalc",      IDM_ABOUT
    END
    SUBMENU "~Edit",                IDM_EDIT
    BEGIN
        MENUITEM "~Copy\tDel",        IDM_COPY
        MENUITEM "~Paste\tIns",       IDM_PASTE
    END
    SUBMENU "F~uncs",                IDM_FUNC /* */
    BEGIN
```

Code Snippet – hpcalc.rc

```
MENUITEM "~DEC-HEX-OCT ...",   IDM_HEX_OCT
SUBMENU "~Hyperbolics",       IDM_HYP
BEGIN
  MENUITEM "~SinH",           IDM_HYP_SIN
  MENUITEM "~CosH",           IDM_HYP_COS
  MENUITEM "~TanH",           IDM_HYP_TAN
  MENUITEM "aSinH",           IDM_HYP_ASIN
  MENUITEM "aCosH",           IDM_HYP_ACOS
  MENUITEM "aTanH",           IDM_HYP_ATAN
END
SUBMENU "~Statistics",        IDM_STAT
BEGIN
  MENUITEM "~X'",             IDM_STAT_X1
  MENUITEM "~Y'",             IDM_STAT_Y1
  MENUITEM "~Slope",          IDM_STAT_SLOPE
  MENUITEM "~Intercept",      IDM_STAT_INT
  MENUITEM "Co~rrelation",    IDM_STAT_CORR
  MENUITEM SEPARATOR
  MENUITEM "pop. S~DEV",       IDM_STAT_SDEV
  MENUITEM "y~Px  Permb",     IDM_STAT_PERM
  MENUITEM "y~Cx  Comb.",     IDM_STAT_COMB
END
SUBMENU "~Constants",         IDM_CON
BEGIN
  MENUITEM "~c\t(m/s)",        IDM_CON_C
  MENUITEM "Planck ~h\t(J s)", IDM_CON_H
  MENUITEM "Charge ~e\t(C)",    IDM_CON_E
  MENUITEM "Permeability æ0\t(H/m)", IDM_CON_U0
  MENUITEM "Permittivity î0\t(F/m)", IDM_CON_E0
  MENUITEM "~Avogadro's #",    IDM_CON_NA
END
SUBMENU "Con~versions",       IDM_CONV
BEGIN
```


Code Snippet – makefile

```
#-----  
#  
# Makefile for HPcalc  
#  
# Using EMX  
#-----  
  
B=/emx/bin/  
I=\emx\include  
L=/emx/lib/  
CC=gcc  
#ifdef DEBUG  
CFLAGS=-g -Wall  
LFLAGS=-g  
OBJ=o  
#else  
CFLAGS=-O -Wall -Zsys -Zomf  
LFLAGS=-s -Zsys -Zomf -Zmap -Zlinker /map  
OBJ=obj  
#endif  
  
A = HPcalc  
VER = 098  
OBJS = $(A).$(OBJ) interface.$(OBJ) change.$(OBJ) ini.$(OBJ) \  
        hex-oct.$(OBJ)  
DIST = $(A).exe $(DIST_SRC)  
DIST_SRC = ReadMe file_id.diz readme-latest.txt readme-old.txt COPYING  
SRC = $(DIST_SRC) interface.c change.c change.h HPcllc$(VER).txt\  
        $(A).h buttons.h interface.h Makefile $(A).ico $(A).rc \  
        ini.c ini.h hex-oct.c hex-oct.h $(A)D.dlg interface-vars.h \  
        ReadMe.Int HPcalc.def HPcalc.c
```

Code Snippet – makefile

```
all: $(A).exe $(A)D.res

$(A).res: $(A).rc $(A).h $(A).ico $(A)D.dlg
    rc -r -i $(I) $(A).rc

$(A)D.res: $(A).res
    copy $(A).res $(A)D.res

$(A).exe: $(OBSJ) $(A)D.res
    $(CC) $(LFLAGS) $(OBSJ) $(A).def $(A).res
    @if "$(DEBUG)" == "" lxlite -B- $@
    @if not "$(DEBUG)" == "" echo Built debug version

$(A).$(OBJ): $(A).c $(A).h interface.h buttons.h change.h ini.h
    $(CC) $(CFLAGS) -c $(A).c

interface.$(OBJ): interface.c interface.h interface-vars.h $(A).h hex-oct.h
    $(CC) $(CFLAGS) -c interface.c

change.$(OBJ): change.c change.h $(A).h interface.h
    $(CC) $(CFLAGS) -c change.c

hex-oct.$(OBJ): hex-oct.c $(A).h interface.h interface-vars.h
    $(CC) $(CFLAGS) -c hex-oct.c

ini.$(OBJ): ini.c ini.h interface.h interface-vars.h change.h
    $(CC) $(CFLAGS) -c ini.c

clean:
    del *.$(OBJ) core rc0* *.res

zip: HPcllc$(VER).zip
```

Code Snippet – makefile

```
HPcllc$(VER).zip: $(DIST) source
    zip -ujo9 $@ $(DIST) HPcllc$(VER)_src.zip
    unzip -t $@
```

```
source: HPcllc$(VER)_src.zip
```

```
HPcllc$(VER)_src.zip: $(SRC)
    zip -ujo9 $@ $(SRC)
    unzip -t $@
```

Working on a “Large” Project

Divide and conquer

Farm out the parts

Keep the parts organized

Test, test, test

Put it all together

Examples: Writing a book/Writing a Program

On Writing

Marcus Fabius Quintilianus
Roman Poet
Circa 65 A.D.

<http://www.scifi-az.com/pdf2/onwrite.pdf>

Writing a Book

Start with an idea

Perceive a need

Imagine a solution

Writing a Program

Start with an idea

Perceive a need

Imagine a solution

Writing a Book

Start with an idea

Perceive a need
Imagine a solution

Organize

Chapters

Sections

Paragraphs

Sentences

Writing a Program

Start with an idea

Perceive a need
Imagine a solution

Organize

Systems

Sub-systems

Routines

Statements

Writing a Book

Start with an idea

Perceive a need
Imagine a solution

Organize

Chapters
Sections
Paragraphs
Sentences

Process the words

Write
Edit

Writing a Program

Start with an idea

Perceive a need
Imagine a solution

Organize

Systems
 Sub-systems
 Routines
 Statements

Process the tokens

Edit
Compile/Debug

Writing a Book

Start with an idea

Perceive a need
Imagine a solution

Organize

Chapters
Sections
Paragraphs
Sentences

Process the words

Write
Edit

Useful Downtime

Starring at the ceiling
Making coffee
Waiting for Godot

Writing a Program

Start with an idea

Perceive a need
Imagine a solution

Organize

Systems
 Sub-systems
 Routines
 Statements

Process the tokens

Edit
Compile/Debug

Useful Downtime

Starring at the whiteboard
Raiding the vending machines
Waiting for the build

Writing a Book

Start with an idea

Perceive a need
Imagine a solution

Organize

Chapters
Sections
Paragraphs
Sentences

Process the words

Write
Edit

Productivity Killers

Starring at the ceiling
Making coffee
Waiting for Godot

Writing a Program

Start with an idea

Perceive a need
Imagine a solution

Organize

Systems
 Sub-systems
 Routines
 Statements

Process the tokens

Edit
Compile/Debug

Productivity Killers

Starring at the whiteboard
Raiding the vending machines
Waiting for the build

The Development Environment

EMX 0.9d

32-bit development environment for OS/2 and DOS
Simplifies porting Unix software to OS/2 and DOS
Creates `native' OS/2 programs and PM applications

Utilities

C/C++ Compiler

Assembler

Debugger

Linking loader

The Development Environment (Continued)

IBM Developer Toolkit

Dialog editor

Icon editor

Font editor

Resource compiler

GNU Utilities

Editors (vim, emacs, etc.)

Text manipulation tools (grep, sed, awk, etc.)

Installing Development Software

Real install program (Warpln)

ZIP file with precompiled binaries

ZIP file of patched sources for compilation

ZIP file of untested sources for compilation

Building the Application

Compile

Resolve dependencies

Link the binary files into an executable

Demonstration

Install EMX

 Compile sample program

 Fix EMX install

Build HPCalc

Install additional software

Build HPCalc

Install additional software

Build HPCalc *and* run it!