The Evolution of a Programmer

Vložil/a cm3l1k1 [1], 13 Květen, 2006 - 00:20

- Fun [2]
- Programming [3]

Velice povedený text znázorňující postup různých typů programátorů k vypsání známé hlášky Hello World. Určitě stojí za přečtení ;o)

High School/Jr.High

```
10 PRINT "HELLO WORLD"
20 END
```

First year in College

```
program Hello(input, output)
  begin
    writeln('Hello World')
  end.
```

Senior year in College

```
(defun hello
  (print
    (cons 'Hello (list 'World))))
```

New professional

```
#include <stdio.h>
void main(void)
{
  char *message[] = {"Hello ", "World"};
  int i;

  for(i = 0; i < 2; ++i)
    printf("%s", message[i]);
  printf("\n");
}
```

Seasoned professional

```
#include <iostream.h>
#include <string.h>

class string
{
private:
  int size;
  char *ptr;

public:
  string() : size(0), ptr(new char('\0')) {}
```

```
  string(const string &amp;s) : size(s.size)
  {
    ptr = new char[size + 1];
    strcpy(ptr, s.ptr);
  }

  ~string()
  {
    delete [] ptr;
  }

  friend ostream &amp;operator <<(ostream &amp;, const string &amp;);
  string &amp;operator=(const char *);
};

ostream &amp;operator<<(ostream &amp;stream, const string &amp;s)
{
  return(stream << s.ptr);
}

string &amp;string::operator=(const char *chrs)
{
  if (this != &amp;chrs)
  {
    delete [] ptr;
   size = strlen(chrs);
    ptr = new char[size + 1];
    strcpy(ptr, chrs);
  }
  return(*this);
}

int main()
{
  string str;

  str = "Hello World";
  cout << str << endl;

  return(0);
}
```

Master Programmer

```
[
uuid(2573F8F4-CFEE-101A-9A9F-00AA00342820)
]
library LHello
{
    // bring in the master library
    importlib("actimp.tlb");
    importlib("actexp.tlb");

    // bring in my interfaces
    #include "pshlo.idl"

    [
    uuid(2573F8F5-CFEE-101A-9A9F-00AA00342820)
```

```
    ]
    cotype THello
 {
 interface IHello;
 interface IPersistFile;
 };
};

[
exe,
uuid(2573F890-CFEE-101A-9A9F-00AA00342820)
]
module CHelloLib
{

    // some code related header files
    importheader(<windows.h>);
    importheader(<ole2.h>);
    importheader(<except.hxx>);
    importheader("pshlo.h");
    importheader("shlo.hxx");
    importheader("mycls.hxx");

    // needed typelibs
    importlib("actimp.tlb");
    importlib("actexp.tlb");
    importlib("thlo.tlb");

    [
    uuid(2573F891-CFEE-101A-9A9F-00AA00342820),
    aggregatable
    ]
    coclass CHello
 {
 cotype THello;
 };
};

#include "ipfix.hxx"

extern HANDLE hEvent;

class CHello : public CHelloBase
{
public:
    IPFIX(CLSID_CHello);

    CHello(IUnknown *pUnk);
    ~CHello();

    HRESULT  __stdcall PrintSz(LPWSTR pwszString);

private:
    static int cObjRef;
};

#include <windows.h>
#include <ole2.h>
#include <stdio.h>
```

```
#include <stdlib.h>
#include "thlo.h"
#include "pshlo.h"
#include "shlo.hxx"
#include "mycls.hxx"

int CHello::cObjRef = 0;

CHello::CHello(IUnknown *pUnk) : CHelloBase(pUnk)
{
    cObjRef++;
    return;
}

HRESULT  __stdcall  CHello::PrintSz(LPWSTR pwszString)
{
    printf("%ws\n", pwszString);
    return(ResultFromScode(S_OK));
}

CHello::~CHello(void)
{

// when the object count goes to zero, stop the server
cObjRef--;
if( cObjRef == 0 )
    PulseEvent(hEvent);

return;
}

#include <windows.h>
#include <ole2.h>
#include "pshlo.h"
#include "shlo.hxx"
#include "mycls.hxx"

HANDLE hEvent;

 int _cdecl main(
int argc,
char * argv[]
) {
ULONG ulRef;
DWORD dwRegistration;
CHelloCF *pCF = new CHelloCF();

hEvent = CreateEvent(NULL, FALSE, FALSE, NULL);

// Initialize the OLE libraries
CoInitializeEx(NULL, COINIT_MULTITHREADED);

CoRegisterClassObject(CLSID_CHello, pCF, CLSCTX_LOCAL_SERVER,
    REGCLS_MULTIPLEUSE, &amp;dwRegistration);

// wait on an event to stop
WaitForSingleObject(hEvent, INFINITE);

// revoke and release the class object
```

```
    CoRevokeClassObject(dwRegistration);
    ulRef = pCF->Release();

    // Tell OLE we are going away.
    CoUninitialize();

    return(0); }

    extern CLSID CLSID_CHello;
    extern UUID LIBID_CHelloLib;

    CLSID CLSID_CHello = { /* 2573F891-CFEE-101A-9A9F-00AA00342820 */
        0x2573F891,
        0xCFEE,
        0x101A,
        { 0x9A, 0x9F, 0x00, 0xAA, 0x00, 0x34, 0x28, 0x20 }
    };

    UUID LIBID_CHelloLib = { /* 2573F890-CFEE-101A-9A9F-00AA00342820 */
        0x2573F890,
        0xCFEE,
        0x101A,
        { 0x9A, 0x9F, 0x00, 0xAA, 0x00, 0x34, 0x28, 0x20 }
    };

    #include <windows.h>
    #include <ole2.h>
    #include <stdlib.h>
    #include <string.h>
    #include <stdio.h>
    #include "pshlo.h"
    #include "shlo.hxx"
    #include "clsid.h"

    int _cdecl main(
    int argc,
    char * argv[]
    ) {
    HRESULT  hRslt;
    IHello        *pHello;
    ULONG  ulCnt;
    IMoniker * pmk;
    WCHAR  wcsT[_MAX_PATH];
    WCHAR  wcsPath[2 * _MAX_PATH];

    // get object path
    wcsPath[0] = '\0';
    wcsT[0] = '\0';
    if( argc  1) {
        mbstowcs(wcsPath, argv[1], strlen(argv[1]) + 1);
        wcsupr(wcsPath);
        }
    else {
        fprintf(stderr, "Object path must be specified\n");
        return(1);
        }

    // get print string
    if(argc  2)
```

```
    mbstowcs(wcsT, argv[2], strlen(argv[2]) + 1);
else
    wcscpy(wcsT, L"Hello World");

printf("Linking to object %ws\n", wcsPath);
printf("Text String %ws\n", wcsT);

// Initialize the OLE libraries
hRslt = CoInitializeEx(NULL, COINIT_MULTITHREADED);

if(SUCCEEDED(hRslt)) {

    hRslt = CreateFileMoniker(wcsPath, &amp;pmk);
    if(SUCCEEDED(hRslt))
 hRslt = BindMoniker(pmk, 0, IID_IHello, (void **)&amp;pHello);

    if(SUCCEEDED(hRslt)) {

 // print a string out
 pHello->PrintSz(wcsT);

 Sleep(2000);
 ulCnt = pHello->Release();
 }
    else
 printf("Failure to connect, status: %lx", hRslt);

    // Tell OLE we are going away.
    CoUninitialize();
    }

return(0);
}
```

Apprentice Hacker

```
#!/usr/local/bin/perl
$msg="Hello, world.\n";
if ($#ARGV >= 0) {
  while(defined($arg=shift(@ARGV))) {
    $outfilename = $arg;
    open(FILE, ">" . $outfilename) || die "Can't write $arg: $!\n";
    print (FILE $msg);
    close(FILE) || die "Can't close $arg: $!\n";
  }
} else {
  print ($msg);
}
1;
```

Experienced Hacker

```
#include <stdio.h>
#define S "Hello, World\n"
main(){exit(printf(S) == strlen(S) ? 0 : 1);}
```

Seasoned Hacker

```
% cc -o a.out ~/src/misc/hw/hw.c
% a.out
```

Guru Hacker

```
% cat
Hello, world.
^D
```

New Manager

```
10 PRINT "HELLO WORLD"
20 END
```

Middle Manager

```
mail -s "Hello, world." bob@b12
Bob, could you please write me a program that prints "Hello, world."?
I need it by tomorrow.
^D
```

Senior Manager

```
% zmail jim
I need a "Hello, world." program by this afternoon.
```

Chief Executive

```
% letter
letter: Command not found.
% mail
To: ^X ^F ^C
% help mail
help: Command not found.

% damn!
!: Event unrecognized
% logout
```

Research Scientist

```
      PROGRAM HELLO
       PRINT *, 'Hello World'
         END
```

Older research Scientist

```
       WRITE (6, 100)
  100 FORMAT (1H ,11HHELLO WORLD)
       CALL EXIT
       END
```

Zdroj: http://www.gnu.org/fun/jokes/helloworld.html [4]

URL článku: https://security-portal.cz/clanky/evolution-programmer

Odkazy:
[1] https://security-portal.cz/users/cm3l1k1
[2] https://security-portal.cz/category/tagy/fun
[3] https://security-portal.cz/category/tagy/programming
[4] http://www.gnu.org/fun/jokes/helloworld.html