



evropský  
sociální  
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,  
MLÁDEŽE A TĚLOVÝCHOVY



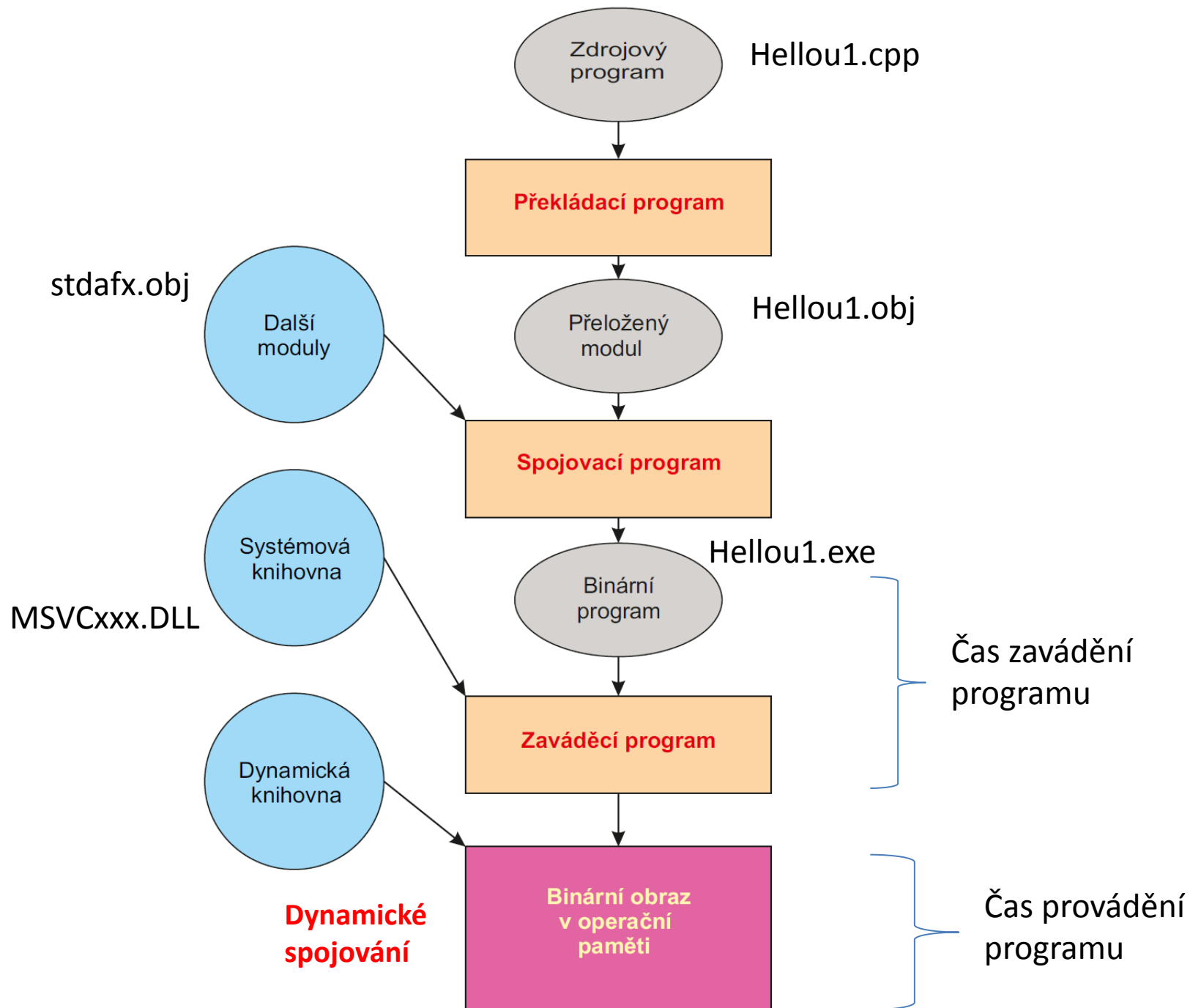
OP Vzdělávání  
pro konkurenceschopnost

## INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

<b>Název školy</b>	Střední průmyslová škola elektrotechnická, Havířov, Příspěvková organizace, Makarenkova 1, Havířov
<b>Název a číslo OP</b>	OP Vzdělávání pro konkurenceschopnost, CZ.1.5
<b>Název projektu</b> <b>Registrační číslo</b>	Podpora odborných kompetencí CZ.1.07/1.5.00/34.0946
<b>Název šablony klíčové aktivity:</b>	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
<b>Číslo materiálu (sady)</b> <b>Název sady</b>	VY32_INOVACE_7-09 Operační systémy I
<b>Autor</b>	Ing. Ladislav Opiol
<b>Tématický celek</b> <b>Předmět</b>	Operační systémy – operační paměť 1. část Operační systémy
<b>Ročník</b>	3.ročník SPŠE
<b>Datum tvorby</b> <b>Ověření ve výuce</b>	Listopad 2012 Listopad 2013
<b>Anotace</b>	Prezentace k výkladu o operační paměti, kompilace programu a spuštění programu
<b>Metodický pokyn</b>	Po prezentaci následuje k procvičení další materiál interaktivní elektronický test

# Řízení operační paměti

- Paměť je velké pole dat ( slov), index pole jsou př. adresy od 0x00000000 do 0xFFFFFFFF, viz. Paměťová mapa
- Paměť je naplněna strojovými kódy programu a sledem adres ( není zde vidět způsob adresace)
- přeložený program je na disku v binárním souboru
- OS umožní zavést kód do paměti na libovolnou adresu
- Zdroj. program vyjadřuje **symbolicky** adresaci př. **i, pom**
- **Kompilátor** vytváří **relativní adresy** ( počet adres od začátku modulu)
- Spojování modulů a převod relativních adres na **absolutní adresy** může nastat ve více místech. Obvykle je provádí **linkér**



File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help

Debug Win32 0

Hex

Process: [0x177C] Hellou1.exe Thread: [0x1378] Main Thread Stack Frame: Hellou1.exe!wmain(int argc, wchar\_t\*

Solution Explorer Disassembly Hellou1.cpp Internet Explorer c...display the webpage

Solution 'Hellou1' (1 project)

- Hellou1
  - External Dependencies
  - Header Files
    - stdafx.h
    - targetver.h
  - Resource Files
  - Source Files
    - Hellou1.cpp
    - stdafx.cpp
    - ReadMe.txt

(Global Scope)

\_tmain(int argc, \_TCHAR\* argv[])

```
// Hellou1.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "stdio.h"

int _tmain(int argc, _TCHAR* argv[])
{
    char i=0;
    printf(" Helloou .....");
    for (i=0; i<5;i++ )
    printf( " i= %d",i);
    return 0;
}
```

100 %

Watch 1

Name	Value	Type
i	0x02 ' '	char

Breakpoints

Name	Labels	Condition	Hit Count
------	--------	-----------	-----------

Microsoft Visual Studio (Administrator) - Hellou1 (Debugging)

File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help

Debug Win32 0

Process: [0x177C] Hellou1.exe Thread: [0x1378] Main Thread Stack Frame: Hellou1.exe!wmain(int argc, wchar\_t\*\* ar

Solution Explorer

- Solution 'Hellou1' (1 project)
  - Hellou1
    - External Dependencies
    - Header Files
      - stdafx.h
      - targetver.h
    - Resource Files
    - Source Files
      - Hellou1.cpp
      - stdafx.cpp
      - ReadMe.txt

Disassembly Hellou1.cpp

Address: wmain(int, wchar\_t\*\*)

Viewing Options

```
008E1797 mov     eax,0CCCCCCCCh
008E179C rep stos dword ptr es:[edi]
char i=0;
008E179E mov     byte ptr [i],0
printf(" Helloou .....");
008E17A2 mov     esi,esp
008E17A4 push   offset string " Helloou ..... " (8E57A8h)
008E17A9 call   dword ptr [__imp__printf (8E82ACh)]
008E17AF add     esp,4
008E17B2 cmp     esi,esp
008E17B4 call   @ILT+435(__RTC_CheckEsp) (8E11B8h)
for (i=0; i<5;i++ )
008E17B9 mov     byte ptr [i],0
008E17BD jmp     wmain+47h (8E17C7h)
008E17BF mov     al,byte ptr [i]
008E17C2 add     al,1
008E17C4 mov     byte ptr [i],al
008E17C7 movsx   eax,byte ptr [i]
008E17CB cmp     eax,5
008E17CE jge     wmain+6Eh (8E17EEh)
printf(" i= %d",i);
```

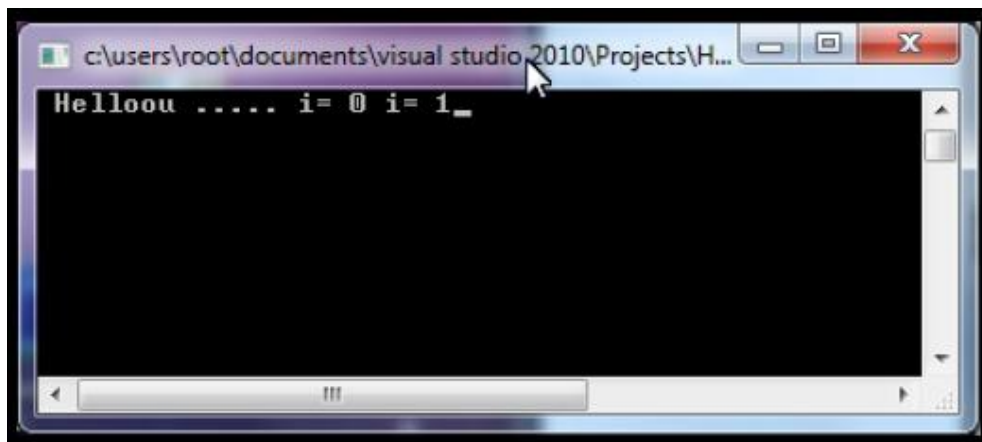
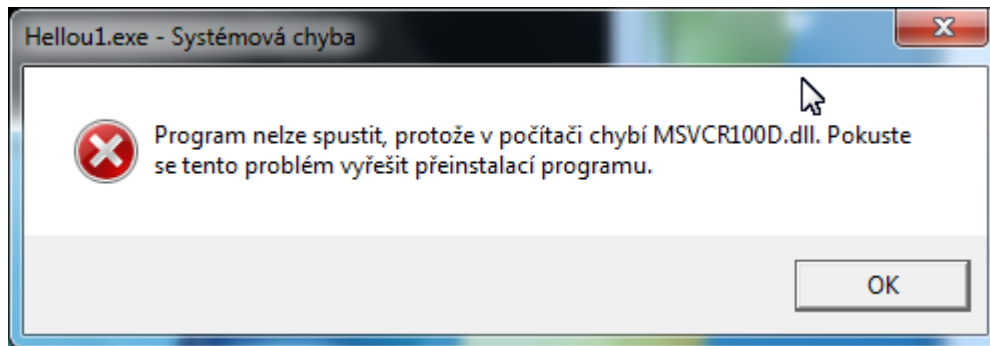
Watch

Name	Value	Type
i	0x02	char

Breakpoints

Name	Labels	Condition	Hit Count
Hellou1.cpp, line 12		(no condition)	break always (currently

# Etapy vzniku a spuštění programu



## Zdroje :

ABRAHAM SILBERSCHATZ, Peter B    z ang. přeložil. Zdzisław PŁOSKI.  
*Podstawy systemów operacyjnych*. Wyd. 4. Warszawa: Wydawnictwa Naukowo-Techniczne, 2001. ISBN 83-204-2689-8.

STALLINGS, William    z ang. přeložil Zbigniew SZALBOT.  
*Systemy operacyjne: struktura i zasady budowy*. Warszawa: Wydawnictwo Naukowe PWN, 2006. ISBN 83-011-4912-4.