



香港中文大學

The Chinese University of Hong Kong

CSCI2510 Computer Organization

Tutorial 02: MASM Basics

Yuhong LIANG

yhliang@cse.cuhk.edu.hk





- You can use the video demo for Tut1 to revisit how to setup basic MASM environment step by step
- Please create a new file for each program. If you want to have two assembly file in one project, please exclude all assembly files until only one included, and rebuild it.
 - [Right Click] the .asm file > [Exclude From Project].
 - You can include it again in [Project] > [Show All Files]

Program structure



.386

.model flat, stdcall

option casemap:none

include windows.inc

include kernel32.inc

include user32.inc

Assembler Directives

.data

MsgCaption db "CSCI2510 Tutorial", 0

MsgBoxText db "Hello, World!", 0

Data Segment

.code

start:

 invoke MessageBox, NULL,addr MsgBoxText, addr MsgCaption, MB_OK

 invoke ExitProcess,NULL

end start

Code Segement

Assembler Directives



- Telling the assembler what to do:
 - Option, configuration, syntax etc...
- `.386`
 - Use 80386 instruction set (intel 1985's architecture, most common supported)
- `.model flat`
 - Memory model of the assembly program
 - Only flat model is supported under Win32 program
- `(.model) stdcall`
 - Function calling convention, parameter passed from right to left (stack)

Includes Files



- casemap:none
 - The assembly language is case insensitive
 - i.e. Label = label = IAbEI
- include windows.inc
- include kerner32.inc
 - Include the files, which handles the system calls
 - E.g. invoke ExitProcess, 0 (valid after include files)
- include user32.inc
 - Graphical User Interface (GUI) elements in windows

Data Segment & Datatypes



- “.data” is also assembler directives
 - Declare and apply some memory space in primary memory (e.g. RAM)

```
.data  Variable name  Datatype  values
MsgCaption db "CSCI2510 Tutorial", 0
MsgBoxText db "Hello, World!", 0
```

- Identifier: variable name
 - a programmer-choice name
 - The first character must be a letter (A..Z, a..z), underscore (), @, ?, or \$. Subsequent character can be digits (0...9).
 - An identifier cannot be the same as an assembler reserved word. Examples: include, option...

Data Segment & Datatypes



number DW 12

number DW 12345

.data

MsgCaption db "CSCI2510 Tutorial", 0

MsgBoxText db "Hello, World!", 0

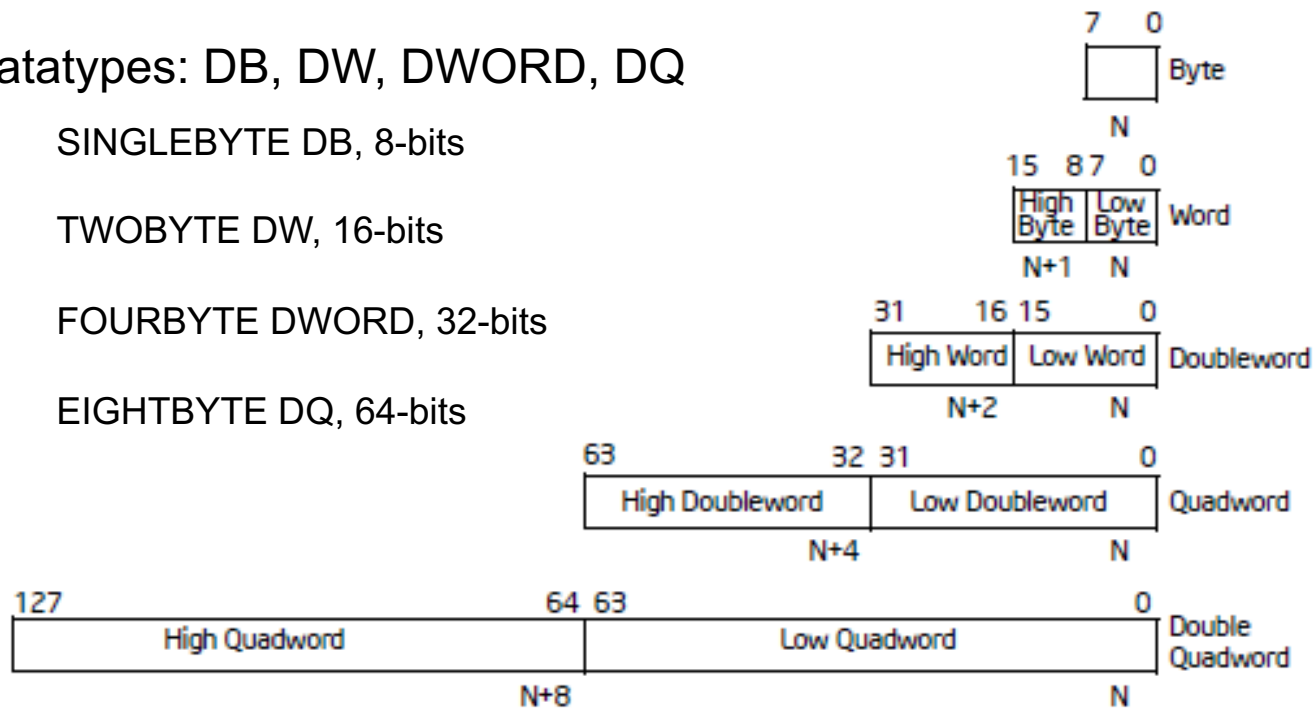
Datatypes: DB, DW, DWORD, DQ

SINGLEBYTE DB, 8-bits

TWOBYTE DW, 16-bits

FOURBYTE DWORD, 32-bits

EIGHTBYTE DQ, 64-bits



Extended ASCII



.data

MsgCaption db "CSCI2510 Tutorial", 0

MsgBoxText db "Hello, World!", 0

← 13 chracters, 67,83,...

ASCII control characters			ASCII printable characters			Extended ASCII characters		
00	NULL	(Null character)	32	space	64	@	96	`
01	SOH	(Start of Header)	33	!	65	A	97	a
02	STX	(Start of Text)	34	"	66	B	98	b
03	ETX	(End of Text)	35	#	67	C	99	c
04	EOT	(End of Trans.)	36	\$	68	D	100	d
05	ENQ	(Enquiry)	37	%	69	E	101	e
06	ACK	(Acknowledgement)	38	&	70	F	102	f
07	BEL	(Bell)	39	'	71	G	103	g
08	BS	(Backspace)	40	(72	H	104	h
09	HT	(Horizontal Tab)	41)	73	I	105	i
10	LF	(Line feed)	42	*	74	J	106	j
11	VT	(Vertical Tab)	43	+	75	K	107	k
12	FF	(Form feed)	44	,	76	L	108	l
13	CR	(Carriage return)	45	-	77	M	109	m
14	SO	(Shift Out)	46	.	78	N	110	n
15	SI	(Shift In)	47	/	79	O	111	o
16	DLE	(Data link escape)	48	0	80	P	112	p
17	DC1	(Device control 1)	49	1	81	Q	113	q
18	DC2	(Device control 2)	50	2	82	R	114	r
19	DC3	(Device control 3)	51	3	83	S	115	s
20	DC4	(Device control 4)	52	4	84	T	116	t
21	NAK	(Negative acknowledge)	53	5	85	U	117	u
22	SYN	(Synchronous idle)	54	6	86	V	118	v
23	ETB	(End of trans. block)	55	7	87	W	119	w
24	CAN	(Cancel)	56	8	88	X	120	x
25	EM	(End of medium)	57	9	89	Y	121	y
26	SUB	(Substitute)	58	:	90	Z	122	z
27	ESC	(Escape)	59	;	91	[123	{
28	FS	(File separator)	60	<	92	\	124	
29	GS	(Group separator)	61	=	93]	125	}
30	RS	(Record separator)	62	>	94	^	126	~
31	US	(Unit separator)	63	?	95	_		
127	DEL	(Delete)						
128	Ç		160	á	192	Ł	224	Ó
129	ü		161	í	193	ł	225	ô
130	é		162	ó	194	Ł	226	õ
131	â		163	ú	195	ł	227	ö
132	ä		164	ñ	196	—	228	ø
133	à		165	Ñ	197	†	229	õ
134	å		166	ª	198	ä	230	µ
135	ç		167	º	199	Ä	231	þ
136	ê		168	¿	200	Ĺ	232	ß
137	ë		169	®	201	Œ	233	Ú
138	è		170	™	202	Œ	234	Û
139	ï		171	½	203	Œ	235	Ü
140	î		172	¼	204	Œ	236	Ý
141	ì		173	í	205	=	237	Ÿ
142	Ä		174	«	206	‡	238	—
143	Å		175	»	207	‡	239	·
144	É		176	„	208	ø	240	≡
145	æ		177	„	209	Ð	241	±
146	Æ		178	„	210	È	242	≡
147	ö		179	„	211	È	243	¾
148	ö		180	„	212	È	244	¶
149	ò		181	„	213	ı	245	§
150	û		182	„	214	ı	246	÷
151	ù		183	„	215	ı	247	°
152	ÿ		184	„	216	ı	248	°
153	Ö		185	„	217	ı	249	°
154	Ü		186	„	218	ı	250	°
155	ø		187	„	219	ı	251	°
156	£		188	„	220	ı	252	°
157	ø		189	„	221	ı	253	°
158	×		190	„	222	ı	254	°
159	f		191	„	223	ı	255	nbsp

Code Segment



```
.code  
  
start:  
    invoke MessageBox, NULL,addr MsgBoxText, addr MsgCaption, MB_OK  
    invoke ExitProcess,NULL  
  
end start
```

- “.code” is also assembler directives
 - State the following segment is the program assembly code
- start:
 - Label that indicates where should the program begins
 - End function with “end start”
- Comment in masm: “;”
 - It will directly comment the whole line

Function in MASM



- Prototype (Declaration):

ExitProcess proto uExitCode:dword

↕ ↕ ↕

FunctionName proto [ParameterName]:DataType, ...

- Usage (call the function)

invoke ExitProcess, 0

↕ ↕

invoke FunctionName, Argument, ...

Three Printout Functions



- **MessageBox** (Win32 message box)
 - Include “user32.inc”
- `crt_printf` (c style function)
- `StdOut` (standard print in MASM32)

Declaration:

- `MessageBox` `PROTO hwnd:DWORD, lpText:DWORD, lpCaption:DWORD, uType:DWORD`
- Usage:

```
.data
MsgCaption db "CSCI2510 Tutorial 1", 0
MsgBoxText db "Win 32 Assembly is Great!", 0

.code
start:
    invoke MessageBox, NULL, addr MsgBoxText, addr MsgCaption, MB_OK
    invoke ExitProcess, NULL
end start
```

Three Printout Functions



- **crt_printf** (c style function)
 - Include “msvcrt.inc” and “msvcrt.lib” (for C functions)
 - Declaration: crt_printf PROTO format:dword ...
 - Usage:

```
.data
PrintFormat db "String: %s, Int: %d", 10, 0
String db "Tutorial", 0
Number db 2

.code
start:
    invoke crt_printf, addr PrintFormat, addr String, Number
    invoke ExitProcess, NULL
end start
```

- StdOut (standard print in MASM32)
- MessageBox (Win32 message box)

Three Printout Functions



- **StdOut** (standard print in MASM32)
 - Include “masm32.inc” and “masm32.lib”
 - Declaration: StdOut proto lpszText:dword
 - Usage:

```
.data
Message db "CSCI2510 Tutorial 2", 10, 0

.code
start:
    invoke StdOut, addr Message
    invoke ExitProcess, NULL
end start
```

Crt_printf Example

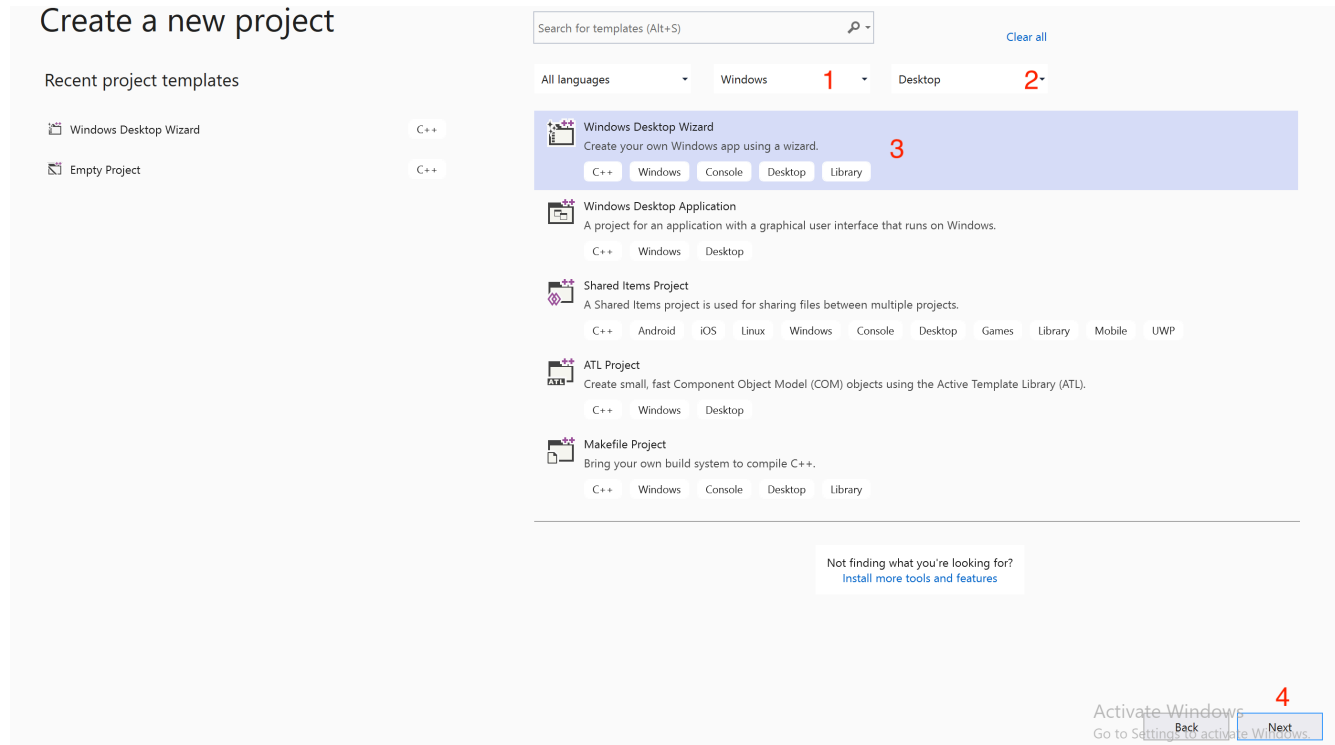


- Download the new library:
 - You can download it at: Blackboard CUHK -> Courses -> CSCI2510 -> Course Content -> Tutorial Notes -> Tut02masm.zip
 - extract the lib and include files to path
C:\Users\yhliang\Documents\masm3\lib
C:\Users\yhliang\Documents\masm3\include

Crt_printf Example



- Open VS community, click [New] > [Project] > [Visual C++] > [Windows Desktop] > [Windows Desktop Wizard] > [Console application] > only choose [Empty project].



Crt_printf Example



- Input the projectname
- Choose empty project

Configure your new project

Windows Desktop Wizard

C++ Windows Console Desktop Library

Project name

tut02

Location

C:\Users\yhliang\source\repos



1

Solution name

tut02

☐ Place solution and project in the same directory

Windows Desktop Project

Application type

Console Application (.exe)

Additional options:

☒ Empty project

1

☐ Precompiled header

☐ Export symbols

☐ MFC headers



Tip: You can also use the Empty Project template to create this kind of project.

2

OK

Cancel

Activate Windows
Go to Settings to activate Windows.

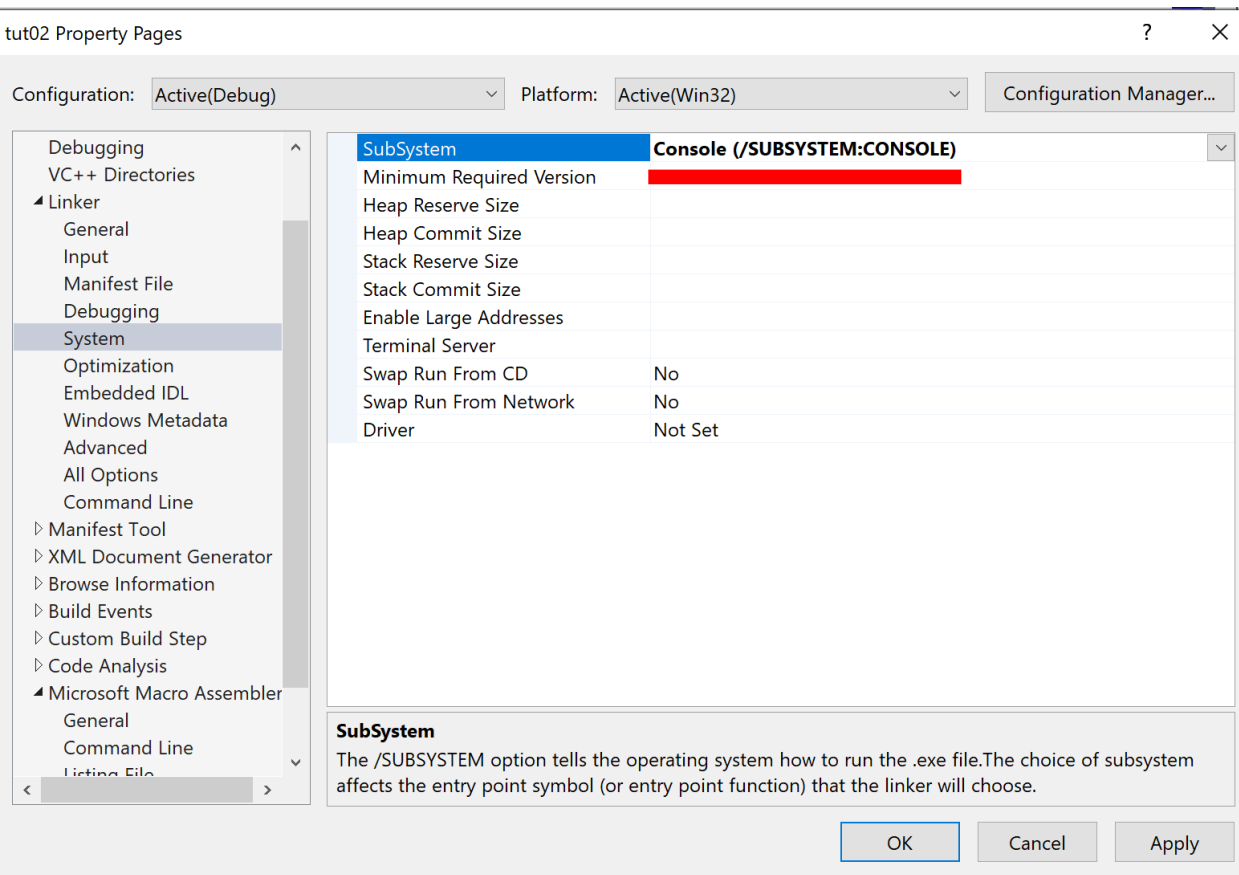
2

Back Create

Crt_printf Example



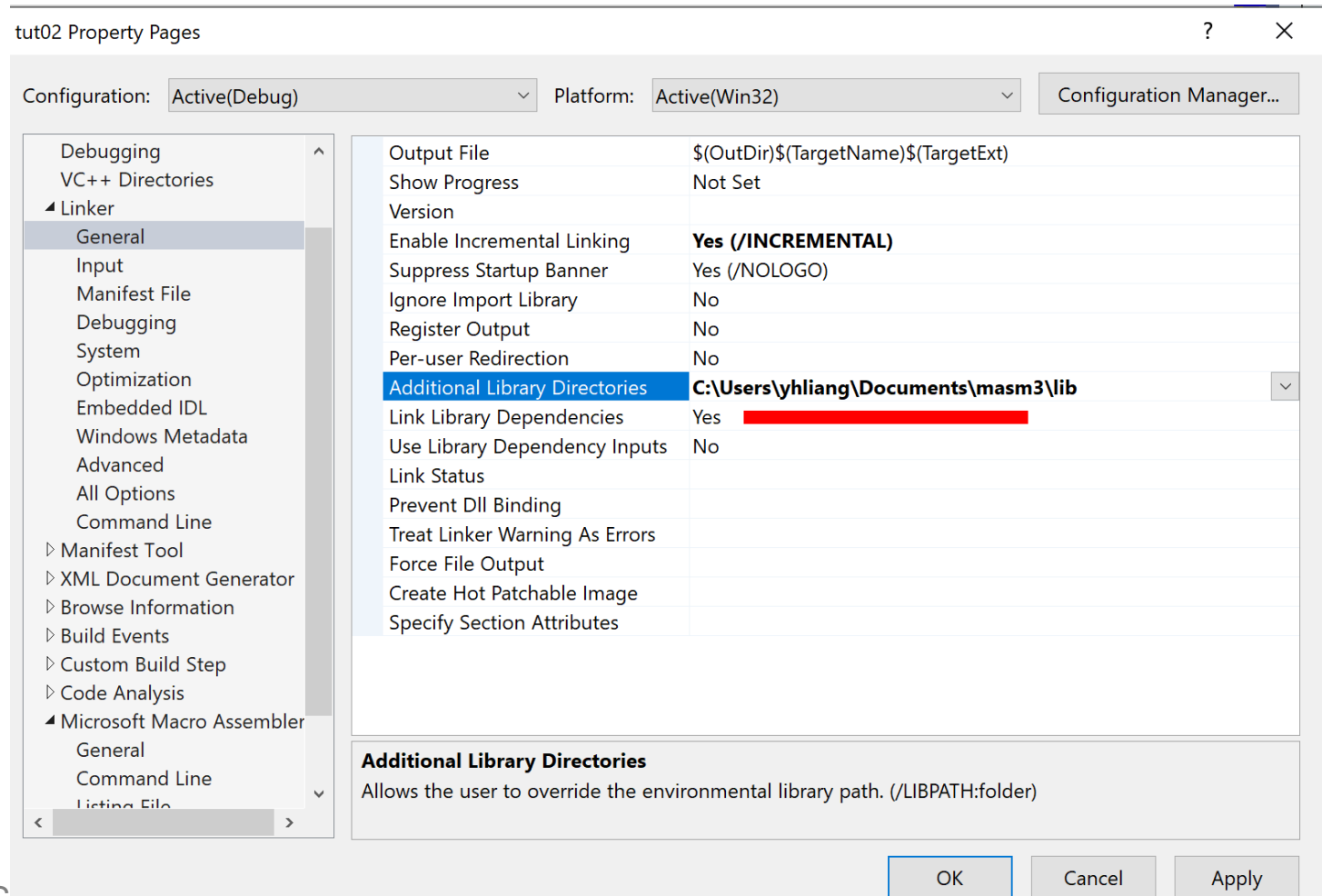
- Same as the steps in page 16 ~ page 19 in TUT01
- Page19 : [Linker] > [System] > [SubSystem]



Crt_printf Example



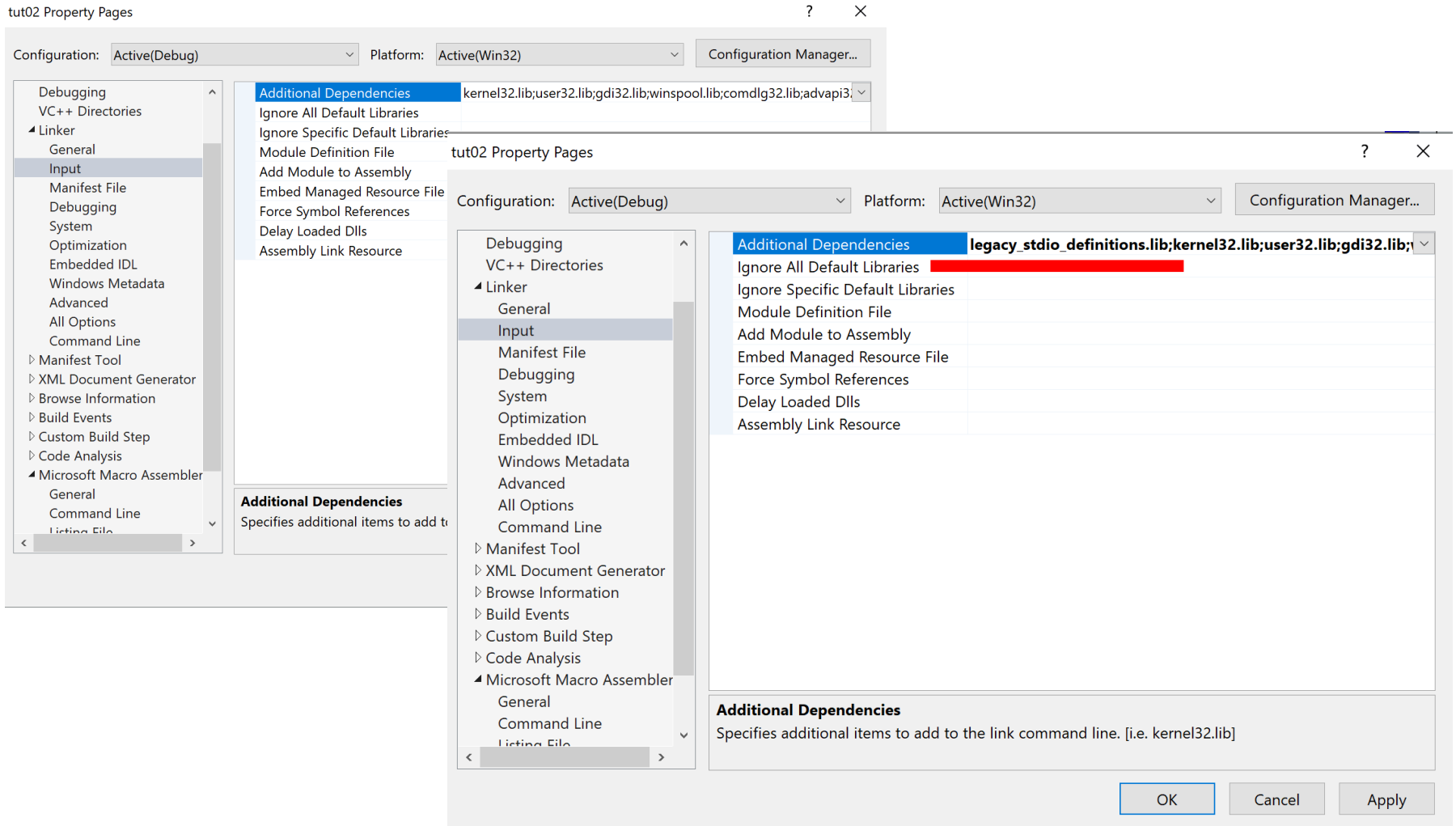
- Page19 : [Linker] > [General] > [Additional Library Directories]



Crt_printf Example



- Page19 : [Linker] > [System] > [Input]>[Additional Dependencies] : legacy_stdio_definitions.lib;



Crt_printf Example



```
.386
.model flat, stdcall
option casemap:none
include windows.inc
include kernel32.inc
include msvcrt.inc
includelib msvcrt.lib
includelib ucrtd.lib

.data
APrompt db "CSCI2510 Tutorial 2", 0

.code
start:
    invoke crt_printf, addr APrompt
    invoke ExitProcess, 0
end start
```

Crt_printf Example



- Build & run

A screenshot of the Microsoft Visual Studio Debug Console window. The title bar reads "Select Microsoft Visual Studio Debug Console". The console output shows "CSCI2510 Tutorial 2" followed by a blank line, then "C:\Users\yhliang\source\repos\tut02\Debug\tut02.exe (process 332) exited with code 0.", and finally "Press any key to close this window . . .". A white cursor is visible on the line "Press any key to close this window . . .".

```
CSCI2510 Tutorial 2

C:\Users\yhliang\source\repos\tut02\Debug\tut02.exe (process 332) exited with code 0.
Press any key to close this window . . .
```



- Program Structure
- Datatypes
- Printout Function