INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

System Software
CSN-252
Introduction

Introduction to system software and machine architecture,	√
A simplified instructional computer,	√
design of one-pass and two-pass assemblers,	√
macro-processor design,	√
loader and linker design,	√
compilers and operating systems,	x
editors, debuggers etc.	?

1. L. L. Beck, System software , 3rd Ed., Pearson Education Asia, 2003.
2. D. M. Dhamdhere, System Programming and Operating System, 2nd Ed., TMH, 2002
3. Peter Abel, IBM PC Assembly Language and Programming, 3rd Ed., PHI, 2000
4. Bryant, R.E. and O'Hallaron, D.R., "Computer Systems: A Programmer's Perspective", Prentice-Hall of India. 2015

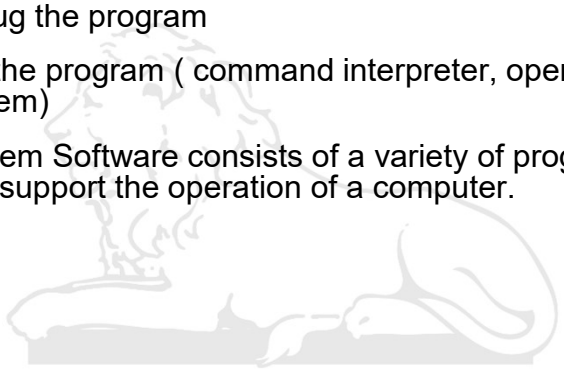
Marks

- CWS 30 (Tut (10) + project (5)
  + surprise Quiz (5) + Quiz (10))
- MTE 30
- ETE 40

Introduction

- ✓ C program – Store in **text file**
- ✓ use a compiler to translate the source file into an executable **object file**
  - ✓ contains four different phases
    - ✓ Preprocessing phase
    - ✓ Compilation phase
    - ✓ Assembly phase
    - ✓ Linking phase
  - ✓ gcc cpp → cc1 → as →ld
  - ✓ C startup code

Introduction



- ✓ debug the program
- ✓ run the program (command interpreter, operating system)
- ✓ System Software consists of a variety of programs that support the operation of a computer.

prog.h

```
int i;
```

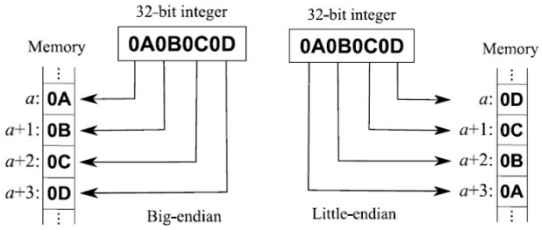
prog.c

```
#include "prog.h"                    size?
int main(){}
```

```
od -x prog.c
```

0000000 6923 636e 756c 6564 2220 7270 676f 682e
0000020 0a22 6e69 2074 616d 6e69 2928 7d7b 000a
0000037

0000000 6923 636e 756c 6564 2220 7270 676f 682e
0000020 0a22 6e69 2074 616d 6e69 2928 7d7b 000a
0000037



ASCII Table

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

IIT ROORKEE

```
gcc –E prog.c

# 1 "prog.c"
# 1 "<built-in>"
# 1 "<command-line>"
# 1 "/usr/include/stdc-predef.h" 1 3 4
# 1 "<command-line>" 2
# 1 "prog.c"
# 1 "prog.h" 1

int i;
# 2 "prog.c" 2

int main(){}
```

gcc –S prog.c

```
.file "prog.c"
      .comm i,4,4
      .text
      .globl main
      .type main, @function
main:
      pushl %ebp
      movl  %esp, %ebp
      popl  %ebp
      ret
      .size main, .-main
      .ident      "GCC:(GNU)4.4.5 20101112(Red Hat 4.4.5-2)"
      .section    .note.GNU-stack,"",@progbits
```

Introduction

- Application software
- System software
- Main differences - machine dependency
- Relationship between system software and machine architecture
- The designer of the assembler must know the instruction formats, addressing modes, etc., of the machine
- Some aspects of the system software do not directly depend on the computing system used
- most real computers have certain characteristics that are unique

IIT ROORKEE

11

Computer Architecture is concerned with the structure and behavior of the computer as seen by the user. It includes the information formats, the instruction set, and techniques for addressing memory. The architectural design of a computer system is concerned with the specifications of the various functional modules, such as processors and memories and structuring them together into a computer system.

Computer organization is concerned with the way the hardware components operate and the way they are connected together to form the computer system.

M. Morris Mano

Computer Architecture refers to those attributes of a system visible to a programmer. Examples of architectural attributes include the instruction set, number of bits used to represent various data types, I/O mechanisms and techniques for addressing memory.

Computer Organization refers to the operational units and their interconnections that realize the architectural specifications. Organizational attributes include those hardware details transparent to programmer, such as control signals; interfaces between the computer and peripherals; and the memory technology used.

William Stallings

Classifying Instruction Set Architectures

- **Design alternatives:**
  - **Type of internal storage in a processor: choices are:**
    - **A stack**
    - **An accumulator**
    - **Set of registers**