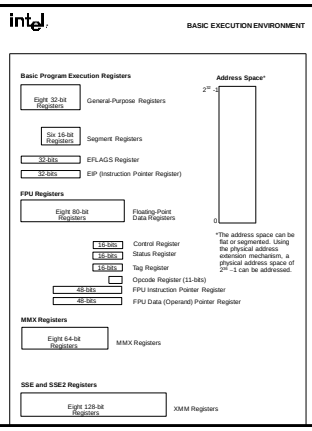
x86 Assembly Language II

CMSC 313
Sections 01, 02

Recap i386 Basic Architecture

- Registers are storage units inside the CPU.
- Registers are much faster than memory.
- 8 General purpose registers in i386:
  - EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP
  - can access subparts of EAX, EBX, ECX and EDX via special names (e.g., EAX→AX→{AH,AL})
- The instruction pointer (EIP) points to (i.e., contains addr of) machine code to be executed next.
- Typically, data moves from memory to registers, is processed, moves from registers back to memory.
- Different addressing modes used.



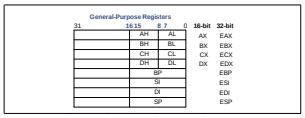


Figure 3-4. Alternate General-Purpose Register Names

4

- EAX—Accumulator for operands and results data.
- EBX—Pointer to data in the DS segment.
- ECX—Counter for string and loop operations.
- EDX—I/O pointer.
- ESI—Pointer to data in the segment pointed to by the DS register; source pointer for string operations.9
- EDI—Pointer to data (or destination) in the segment pointed to by the ES register; destination pointer for string operations.
- ESP—Stack pointer (in the SS segment).
- EBP—Pointer to data on the stack (in the SS segment).

5

toupper.asm

- Use Linux system call to output prompt.
- Use Linux system call to get user input.
- Scan each character of user input and convert all lower case characters to upper case.
- Learn how to:
  - work with 8-bit data
  - specify ASCII constant
  - compare values
  - do loop control
- Use gdb to trace execution

6

[Show source of toupper.asm]

GDB Debugger

Debugging Assembly Language Programs

- Cannot just put print statements everywhere.
- Use gdb to:
  - examine contents of registers
  - examine contents of memory
  - set breakpoints
  - single-step through program
- READ THE GDB SUMMARY ONLINE!

Summary of gdb commands, p1

Command	Example	Description
Run		start program
quit		quit out of gdb
cont		continue execution after a break
break [addr]	break *_start+5	sets a breakpoint
delete [n]	delete 4	removes nth breakpoint
Delete		removes all breakpoints
info break		lists all breakpoints
list _start		list a few lines of the source code around _start
list 7		list 10 lines of the source code starting on line 7
list 7, 20		list lines 7 thru 20 of the source code

10

Summary of gdb commands, p2

Command	Example	Description
Stepi		execute next instruction
stepi [n]	stepi 4	execute next n instructions
Nexti		execute next instruction, stepping over function calls
nexti [n]	nexti 4	execute next n instructions, stepping over function calls
where		show where execution halted
disas [addr]	disas _start	disassemble instructions at given address

11

Summary of gdb commands, p3

Command	Example	Description
info registers		dump contents of all registers
print/d [expr]	print/d $ecx	print expression in decimal
print/x [expr]	print/x $ecx	print expression in hex
print/t [expr]	print/t $ecx	print expression in binary
x/NFU [addr]	x/12xw &msg	Examine contents of memory in given format
display [expr]	display $eax	automatically print the expression each time the program is halted
info display		show list of automatically displays
undisplay [n]	undisplay 1	remove an automatic display

12