CS232: Lab3
Assembly Programming
(MIPS ISA)

Prajeeth.S

Registers in MIPS ISA:

Symbolic Name	Number	Usage
zero	0	Constant 0.
at	1	Reserved for the assembler.
v0 - v1	2 - 3	Result Registers.
a0 - a3	4 - 7	Argument Registers $1 \cdots 4$.
t0 - t9	8 - 15, 24 - 25	Temporary Registers $0 \cdots 9$.
s0 - s7	16 - 23	Saved Registers $0 \cdots 7$.
k0 - k1	26 - 27	Kernel Registers $0 \cdots 1$.
gp	28	Global Data Pointer.
sp	29	Stack Pointer.
fp	30	Frame Pointer.
ra	31	Return Address.

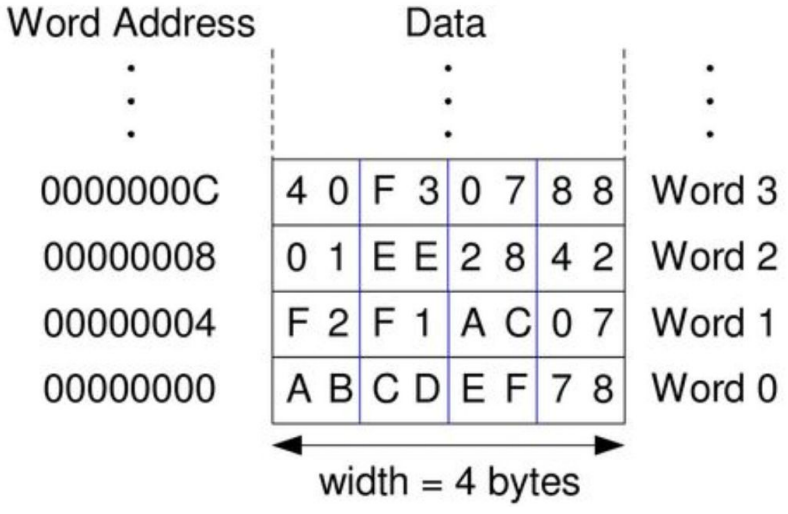
System Calls:

Table of Available Services

Service	Code in $v0	Arguments	Result
print integer	1	$a0 = integer to print	
print float	2	$f12 = float to print	
print double	3	$f12 = double to print	
print string	4	$a0 = address of null-terminated string to print	
read integer	5		$v0 contains integer read
read float	6		$f0 contains float read
read double	7		$f0 contains double read
read string	8	$a0 = address of input buffer $a1 = maximum number of characters to read	*See note below table*
sbrk (allocate heap memory)	9	$a0 = number of bytes to allocate	$v0 contains address of allocated memory

Source: https://courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html

MIPS - Byte-Addressable memory



Word-addressable: Each address refers to a 32-bit data

Byte-addressable: Each address refers to a 32-bit data
Eg: MIPS

Credits: https://safari.ethz.ch/digitaltechnik/spring2021/doku.php?id=start

Loading and Storing Words/Bytes:

- Use lb(load byte), sb(store byte), lw(load word), sw(store word)
- Examples:
  - lw $s3, 4($sp) : Loads word at address $sp+4 into $s3
  - Similarly for store: sw $s3, 4($sp)
  - lb $t0, 20($a0) : Loads one byte at address $a0 + 4
  - …

Credits: https://safari.ethz.ch/digitaltechnik/spring2021/doku.php?id=start

Function Calls:

- Modular Code
- Takes in Arguments and Returns some value
- Two components:
  - **Caller**: The code that calls the function
  - **Callee**: The function that is being called
- Caller: Passes arguments, jumps to callee
- Callee: Performs some procedure, returns the result to caller
  - Must not overwrite registers or memory needed by the caller

Credits: https://safari.ethz.ch/digitaltechnik/spring2021/doku.php?id=start

Function Calls(MIPS ISA):

- Arguments($a0-$a3)
- Return value($v0, $v1)
- Caller: Uses **jal** instruction
- Callee: Uses **jr** instruction

Credits: https://safari.ethz.ch/digitaltechnik/spring2021/doku.php?id=start

Assembler Directives

- Directions to the assembler to take some action or change a setting.
- Assembler directives do not represent instructions, and are not translated into machine code.

Show: https://class.ece.uw.edu/469/peckol/doc/ARM/assemblerDirectives.pdf