Using QtSPIM

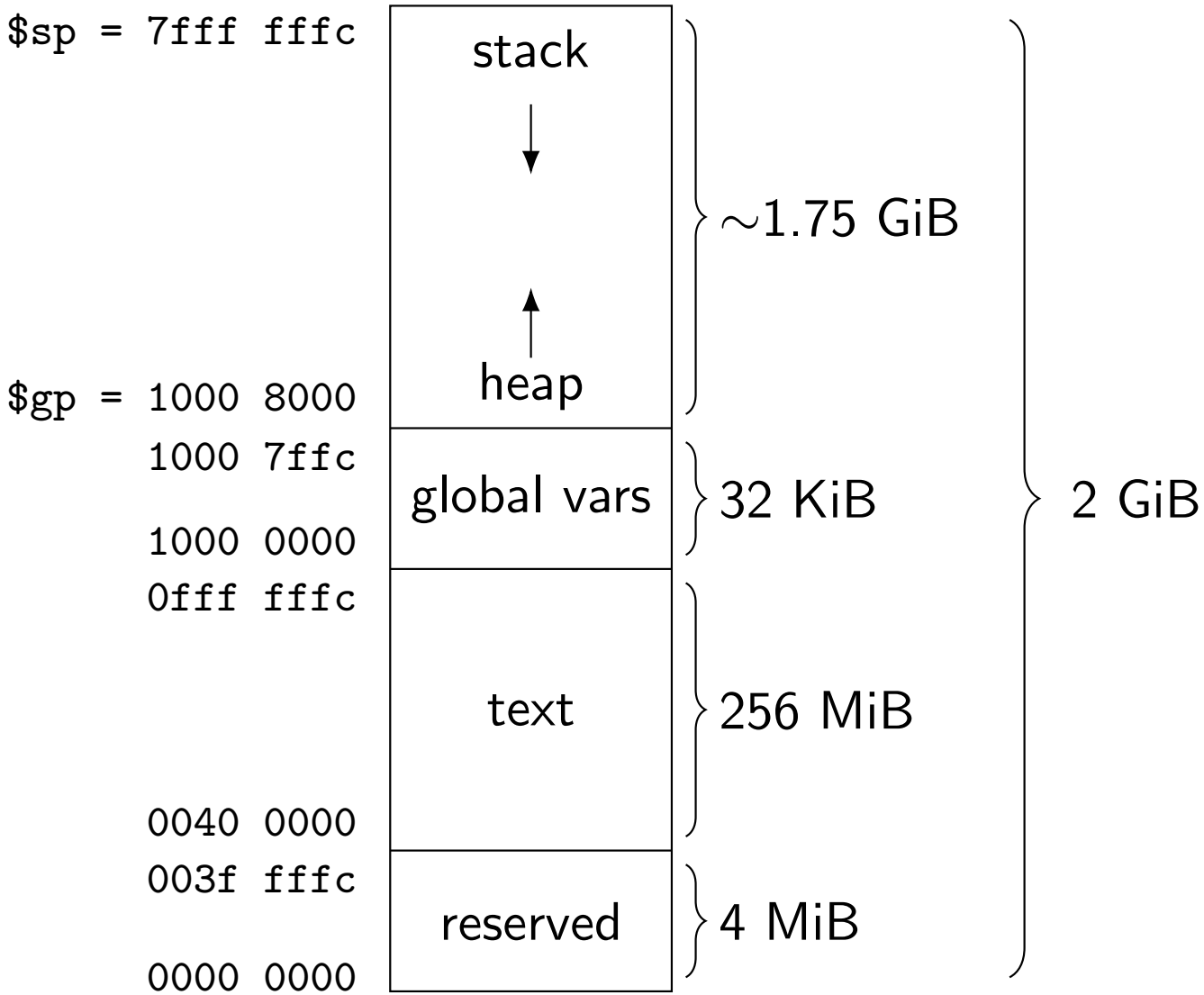
Class 39

QtSPIM

- qtspim is an assembler, simulator, and debugger for MIPS
- it allows you to assemble and run MIPS programs

- in addition to the code we've been learning to write, QtSPIM (like all assemblers) requires the use of some directives
- all directives start with a period e.g., `.data`

Memory


$sp = 7fff fffc
stack
heap
~1.75 GiB
$gp = 1000 8000
1000 7ffc
global vars
32 KiB
1000 0000
0fff fffc
text
256 MiB
0040 0000
003f fffc
reserved
4 MiB
0000 0000
2 GiB

Data

- global variables are stored in RAM
- local variables are stored on the stack
- to declare, and initialilze, a global variable, a program must have a `.data` section, containing any of the following directives

.byte	an 8-bit variable
.word	a 32-bit variable
.ascii	an ASCII string (rarely used)
.asciiz	a NULL-terminated ASCII string
.space	n bytes of uninitialized memory

Data Example

```
   .data
message:
   .asciiz "Hello, world!"

newline:
   .asciiz "\n"

array:
   .word 10, 20, 30
```

Code Section

- program code begins with the `.text` directive
- in addition, the start of the `main` function is denoted with the `.globl` directive
- main is terminated with the `.end` directive

Complete Program

```
1  # print a greeting message on the screen
2
3    .data
4  message:
5    .asciiz "Hello, world!"
6
7  newline:
8    .asciiz "\n"
9
10   .text
11   .globl main
12
13 main:
14   la     $a0, message          # address of message into a0
15   li     $v0, 0x04             # print-string system call
16   syscall
17
18   la     $a0, newline          # address of newline into a0
19   li     $v0, 0x04             # print-string system call
20   syscall
21
22   li     $v0, 0x0a             # exit system call
23   syscall
```

Style

- lowercase hex letters
- labels in column 1
- instructions and directives in column 3
- arguments in column 9
- comments in column 29
- a comment on (almost) every instruction
- a space after every comma
- no camel case

Syscalls

function	**$v0**	**arguments**	**result**
print int	1	$a0 = integer	
print string	4	$a0 = string	
read int	5		integer in $v0
read string	8	$a0 = buffer	
		$a1 = max chars	
exit	10		
print character	11	$a0 = character	
read character	12		character in $v0

Resource

Jorgensen, MIPS Assembly Language Programming using QtSpim

`http://www.egr.unlv.edu/~ed/MIPStextSMv11.pdf`