

Simple Statements

MIPS System Calls

- The call code is placed in register \$v0

Service	Call Code	Arguments	Results
print_int	1	\$a0=integer	
print_string	4	\$a0=address of string	
read_int	5		integer in \$v0
read_string	8	\$a0=buffer \$a1=length	
exit	10		
print_character	11	\$a0=integer	
read_character	12		character in \$v0

Write Statement

```
WriteStatement : WRITESY LEFTPARENSY
                Expression RIGHTPARENSY
                {writestatement($3);}
| | WRITELNSY
  {writeInstatement();}
;
```

- Notice that WriteStatement has no semantic value

Print an Integer

```
li $v0, 1 # System call code for print_int
li $a0, 5 # Integer to print
syscall # Print it
```

Assignment Statement

Assignment : Designator ASSIGNSY Expression
 {assign(\$1, \$3);}
 ;

- We only are dealing with simple variables at this point
- The left hand side has to be something that can be stored to
 - For example, it can't be a constant

Read Statement

ReadStatement : READSY LEFTPARENSY Designator
 RIGHTPARENSY
 {readstatement(\$3);}
 ;

Null Statement

```
NullStatement : /* empty */  
              ;
```

- This is a tough one to implement
- Can you figure it out?