

Lecture 5

Iteration

FIT 1008
Introduction to Computer Science



COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

WARNING

This material has been reproduced and communicated to you by or on behalf of Monash University pursuant to Part VB of the Copyright Act 1968 (the Act). The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice.

Objectives for this lecture

- To understand how they are used to translate **iteration** (loops)
 - while
 - for

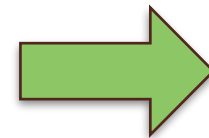
Reminder: Iteration

- Iteration is the repetition of a section of code
 - In Python, with `while`, `for`
 - `while` tests condition before loop entry
 - `for` is a shorthand for `while`
- Achieved by sending control from the end of the loop back to the beginning
 - Test some condition to prevent infinite loop

Iteration: **for**

- A **for** loop is essentially a simpler version of a **while** loop:
 - Initialisation, condition and increment code all in one place
- To translate a **for** loop into MIPS, write it as a **while** loop

```
for i in range(init, cond, inc):  
    body
```



```
i = init  
while (cond):  
    body  
    inc
```

```
for i in range(10):  
    print(i)
```

```
i = 0  
while i < 10:  
    print(i)  
    i += 1
```

```
my_list = [1, 2, 3]  
for element in my_list:  
    print(element)
```

```
my_list = [1, 2, 3]  
i = 0  
while i < len(my_list):  
    print(my_list[i])  
    i += 1
```

```
for i in range(3, 10, 2):  
    print(i)
```

```
i = 3  
while i < 10:  
    print(i)  
    i += 2
```

Before attempting translation of a **for** structure, turn it into **while**

Iteration is truly a special case of **selection**

```
while condition:  
    # do stuff  
  
# stuff after the loop
```

```
loop:  
if condition is not met j endloop:  
    # do stuff  
    j loop  
  
endloop:  
# stuff after the loop
```

factorial.py

```
f = 1

n = int(input("Enter int: "))

while n > 0:

    f = f * n

    n -= 1

print(f)
```


factorial.asm

```
    # set up strings
    # set up n = 0 and f = 1
    # read n

loop:
    # if n <= 0 goto endloop
    # f = f * n
    # n -= 1
    # goto loop

endloop:
    # print f
    # exit
```

Let's look
at this

if $n \leq 0$ goto endloop

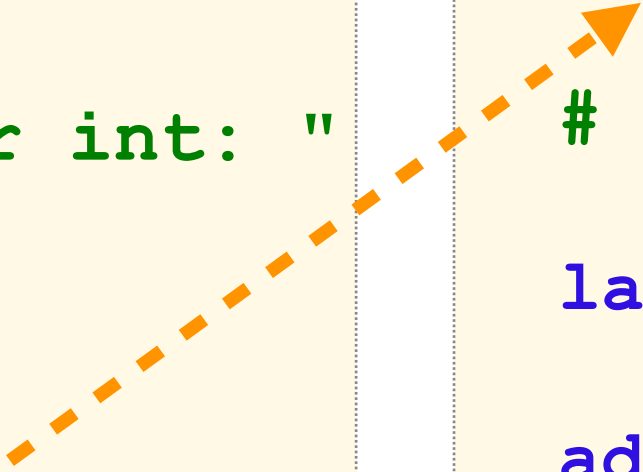
```
lw    $t0, n
```

```
slt    $t1, $0, $t0
```

```
beq    $t1, $0, endloop
```

setup

```
        .data
prompt:  .ascii "Enter int: "
        f:  .word 1
        n:  .word 0
```



```
        .text
# print prompt

la      $a0, prompt

addi    $v0, $0, 4

syscall

# read n

addi    $v0, $0, 5

syscall

sw      $v0, n
```

loop

```
# if n <= 0 goto endloop
```

```
loop:
```

```
    lw    $t0, n
```

```
    slt   $t1, $0, $t0
```

```
    beq   $t1, $0, endloop
```

```
    lw    $t1, f
```

```
    # f = f * n
```

```
    mult  $t1, $t0
```

```
    mflo  $t1
```

```
    sw    $t1, f
```

```
    # n = n - 1
```

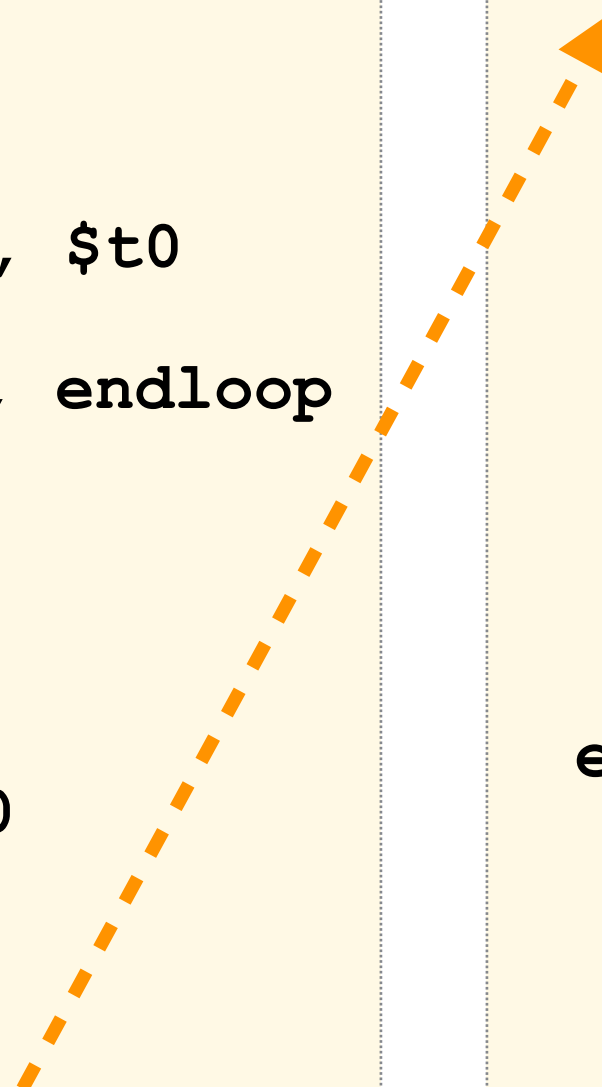
```
    lw    $t0, n
```

```
    addi  $t0, $t0, -1
```

```
    sw    $t0, n
```

```
    j     loop
```

```
endloop:
```



endloop

```
endloop:
```

```
lw      $a0, f          # print f
```

```
addi    $v0, $0, 1
```

```
syscall
```

```
addi    $v0, $0, 10     # exit
```

```
syscall
```

Summary

- MIPS branch and jump instructions
- Selection
 - if-else
- Iteration (loops)
 - while
 - for
- Instruction Format
 - R type
 - I type
 - J type