#8 : MIPS Programming IV

Computer Architecture 2019/2020

João Soares & Ricardo Rocha

Computer Science Department, Faculty of Sciences, University of Porto

Simple Procedure Call

```
int proc (int arg1, int arg2) {   // arguments in $a0 and $a1
   int r = ...;                   // r in $s0, need to save $s0 on stack
   return r;                      // return value in $v0
}
-----------------------------------------------------------------
_main: ...
       li    $a0, ...            # put argument $a0
       li    $a1, ...            # put argument $a1
       jal   _proc               # jump and link
       ...

_proc: addiu $sp, $sp, -4        # adjust stack pointer
       sw    $s0, 0($sp)         # save $s0
       ...                       # return value in $v0
       lw    $s0, 0($sp)         # restore $s0
       addiu $sp, $sp, 4         # restore stack pointer
       jr    $ra                 # return
```

Recursive Procedure Call

```
int proc (int arg1, int arg2) {   // arguments in $a0 and $a1
   ... proc(...) ...;             // recursive call
   return r;                      // return value in $v0
}
----------------------------------------------------------------------
_proc: addiu $sp, $sp, -12       # adjust stack pointer
       sw    $ra, 8($sp)         # save $ra
       sw    $s0, 4($sp)         # save $s0
       ...
       sw    $t0, 0($sp)         # save $t0
       jal   _proc               # recursive call
       ...                       # return value in $v0
       lw    $ra, 8($sp)         # restore $ra
       lw    $s0, 4($sp)         # restore $s0
       addiu $sp, $sp, 12        # restore stack pointer
       jr    $ra                 # return
```