

dúvida da aula passada

tradução de

```
while (save[i] == k)
    i++;
```

i no \$S3

k no \$S5

base de save em \$S6

LOOP: `sl $t1, $S3, 2` (mult. por 4, word=32)

`add $t1, $t1, $S6`

`lw $t0, 0($t1`

\Rightarrow `lw $t0, $S5, EXIT`

`addi $S3, $S3, 1`

`j LOOP`

EXIT:

Para chamador recursivos é necessário guardar o conteúdo dos parâmetros

Ex:

```
int fact (int n) {
```

```
    if (n < 1) return 1;
```

```
    else return (n + fact (n-1));
```

```
}
```

FACT: addi \$sp, \$sp, -8
 mov \$ra, 4(\$sp)
 mov \$a0, 0(\$sp) *
 rti \$t0, \$a0, 1
 beq \$t0, \$zero, L1
 addi \$v0, \$zero, 1
 addi \$sp, \$sp, 8
 jr \$ra

54

L1: addi \$a0, \$a0, -1
 jal fact
 lw \$a0, 0(\$sp)
 lw \$ra, 4(\$sp)
 addi \$sp, \$sp, 8
 mal ~~\$t0~~, \$a0, \$v0
 jr \$ra

→ chamada recursiva

~~rti~~ = SLT = set less than

o mesmo cuidado ~~deve~~ deve ser tomado para funções que chamam funções

É interessante notar que além das variáveis locais

caso tenhamos as variáveis globais. Para o segundo tipo temos mais um registrador o $\$gp$ (global pointer) (35)