

Funkcje i procedury

(Przekazywanie parametrów)

Procedura:

```
void proc_name( args );
```

```
.type proc_name, @function
```

```
proc_name:    ...  
              ...  
              RET
```

Funkcja:

```
ret_type func_name( args );
```

```
.type func_name, @function
```

```
func_name: ...
```

```
...
```

```
MOV ret_value, %rax (eax, edx:eax)
```

```
RET
```

Przekazywanie parametrów i rezultatów:

Parametry:

- ❑ przez rejestry
- ~~❑ przez zmienne globalne~~
- ❑ przez stos

Rezultaty:

- ❑ w rejestrze/rejestrach
- ~~❑ w zmiennych globalnych~~

Rejestry (64):

RAX, RBX, RCX, RDX, RSI, RDI, ...

```
movq $1,%rax
```

osiem dodatkowych rejestrów GPR: R8..R15

osiem dodatkowych rejestrów SSE: xmm8..xmm15

Intel AVX (Advanced Vector eXtensions) – 16
rejestrów (256 bitów): ymm0..ymm15

Szczegółowe informacje:

„System V Application Binary Interface”

<http://www.x86-64.org/documentation/abi.pdf>

- dokumentacja firm INTEL i AMD
- dokumentacja GNU GCC

Przekazywanie parametrów (64):

MEMORY – stack

INTEGER - %rdi, %rsi, %rdx, %rcx, %r8, %r9

SSE - %xmm0, ..., %xmm7

SSEUP - part of %ymm0, ..., %ymm7

X87, X87UP,
COMPLEX_X87 - stack

%al - liczba użytych rejestrów wektorowych (dla funkcji o zmiennej liczbie argumentów, np. printf)

Ochrona rejestrów (64):

`%rsp` – stack pointer

`%rbp, %rbx, %r12, %r13, %r14, %r15`

Zwracanie rezultatu (64):

INTEGER - `%rax` (`, %rdx`)

SSE - `%xmm0` (`, %xmm1`)

X87 - `%st0`

X87COMPLEX - `%st0, %st1`