

x86-64 Cheatsheet

Генерални регистри

RAX, RBX, RCX, RDX, RSI, RDI, RBP, RSP, R8-R15: 64-битни регистри
EAX, AX, AL: Доњих 32/16/8 бита од регистра RAX.
EDI, DI, DIL: Доњих 32/16/8 бита од регистра RDI.
RIP: Показивач инструкција (никада се не користи директно)
RSP: Показивач врха стека
RBP: Показивач дна стек оквира (base pointer)

Приступ меморији

- `MOV RAX, [RBX]`: У регистар RAX упиши 64-битну вредност са адресе из RBX
- `MOV [RBX], EAX`: На адресу из RBX упиши 32-битну вредност из EAX
- `LEA RAX, [RBX+4*RCX]`: У регистар RAX упиши адресу $RBX + 4*RCX$
- Адресирање: $[base + index*scale + offset]$

Аритметика и битовске операције

- `ADD RAX, RBX`: $RAX = RAX + RBX$
- `SUB RAX, RBX`: $RAX = RAX - RBX$
- `IDIV, DIV RDI`: $RAX = RDX:RAX / RDI$
 $RDX = RDX:RAX \% RDI$
- `IMUL, MUL RDI`: $RDX:RAX = RAX * RDI$
- `INC, DEC`: $++ / -$
- `AND, OR, XOR, NOT`: Битовске операције
- `SHL, SHR, SAR, ROL, ROR`: Померања

Заставице (RFLAGS)

- CF: Пренос (unsigned overflow)
- PF: Парност
- ZF: Нула
- SF: Знак ($SF = 1 \Rightarrow$ негативно)
- OF: Прелив (signed overflow)

Постављају: Аритметика, `CMR, TEST`

Скокови и гранање

- | | |
|---|---|
| • <code>JMP label</code> : Безусловни скок | није једнако/није нула |
| • <code>CALL label</code> : Позив функције | • <code>JL, JG, JLE, JGE</code> : Са предзнаком |
| • <code>RET</code> : Повратак из функције | • <code>JB, JA, JBE, JAE</code> : Без предзнака |
| • <code>JE/JZ, JNE/JNZ</code> : Скок ако је једнако/нула, | |

Конвенција за позив функција

- **RDI, RSI, RDX, RCX, R8, R9:** Првих 6 целобројних/показивачких аргумената
- **RAX:** Повратна вредност
- **RAX, RCX, RDX, RSI, RDI, R8-R11:** Чувају се пре позива функције (caller-saved)
- **RBX, RBP, R12-R15:** Чувају се на почетку функције (callee-saved)
- **Стек:** 16-бајтно поравнање пре позива функције

Почетак функције:

ručno:	automatski:
PUSH RBP < >	
MOV RBP, RSP 0	
SUB RSP, <size>	ENTER <size>, 0

Крај функције:

ručno:	automatski:
MOV RSP, RBP	
POP RBP	LEAVE
RET	RET

Компилација и дебаговање

Компилација:

```
g++ file1.cpp file2.s -g -o program
```

Покретање програма:

```
./program
```

Покретање дебагера:

```
gdb ./program
```

Инструкције дебагера:

- **run:** Покрени програм
- **break:** Обележава место за заустављање извршавања
 - **b ime_funkcije:** заустављање на почетку функције ime_funkcije
 - **b 0x400123:** заустављање на адреси 0x400123
 - **b file.cpp:10:** заустављање на линији 10 у фајлу file.cpp
- **next:** Изврши следећу инструкцију, не улази у функције
- **step:** Изврши следећу инструкцију, улази у функције
- **continue:** Настави извршавање до следеће тачке прекида
- **print RAX:** Прикажи вредност регистра RAX
- **backtrace:** Прикажи стек позива функција
- **quit:** Изађи из дебагера
- **tui reg general:** Прикажи регистре и код у TUI режиму