

# 1 Intel (IA-64) のレジスタ

`%rax, %rbx, %rcx, %rdx, %rsi, %rdi, %r8, %r9, %r10, %r11, %r12, %r13, %r14, %r15` が (比較的) 汎用に使える 64 ビットレジスタ。上記以外に、ベースポインタ `%rbp` とスタックポインタ `%rsp` がある。

(`%eax, %ebx, %ecx, %edx, %esi, %edi, %r8d, %r9d, %r10d, %r11d, %r12d, %r13d, %r14d, %r15d` とすると 32 ビットレジスタになるが、この授業では使わない。)

## 2 各種命令 (64 ビット。雰囲気)

(後ろについている `q` を 1 にすると、だいたい 32 ビット命令になる。)

`addq A, B`: `A` と `B` を足した値を `B` に入れる。(C の `B += A`; に相当。)

`subq A, B`: 同様に引き算。( `B -= A`; )

`incq A`: `A` の値に 1 を加える。( `A++`; )

`decq A`: `A` の値から 1 を引く。( `A--`; )

`imulq A, B`: `A` と `B` を掛けた値を `B` に入れる。( `B *= A`; )

`idivq A`: `%rdx:%rax` を `A` で割った商と余りをそれぞれ `%rax`、`%rdx` に入れる。

`sarq A, B`: `B` を `A` ビット、右に符号付きでシフトする。

`negq A`:  $-A$  を `A` に入れる。

`movq A, B`: `A` の値を `B` に入れる。

`cmpq A, B`: `A` と `B` を比べて、フラグをたてる。

`je L`: 等しければ `L` へ飛ぶ。そうでなければ次の命令に行く。

`jg L`: 大きければ `L` へ飛ぶ。そうでなければ次の命令に行く。

`jle L`: 小さければ `L` へ飛ぶ。そうでなければ次の命令に行く。

`jmp L`: 無条件に `L` へ飛ぶ。

`call L`: 戻り番地をスタックにプッシュしてから、`L` に飛ぶ。

`ret`: スタックから戻り番地をポップして、そこに飛ぶ。

`pushq A`: スタックに `A` をプッシュする。

`popq A`: スタックから `A` をポップする。

`nop`: 何もしない。

その他、とても多くの命令がある。