

## Conjunto de instruções mais frequentes na arquitectura IA32

Instrução	Operações (em C)	Descrição	↔
mov D, F	D = F;	Copia informação	
push F	esp -= 4; *esp = F;	Empilha no stack	
pop Dr	Dr = *esp; esp += 4;	Desempilha do stack	
lea Dr, Fm	Dr = &Fm;	Copia endereço de memória	
movsx Dr, Fv	Dr = Fv; sizeof(Dr)>sizeof(Fv)	Copia estendendo o sinal	
movzx Dr, Fv	Dr = Fv; sizeof(Dr)>sizeof(Fv)	Copia estendendo com zeros	
inc D	D += 1; (não actualiza cf e of)	Incrementa	
dec D	D -= 1; (não actualiza cf e of)	Decrementa	
neg D	D = -D;	Nega	
add D, F	D += F;	Adiciona	
sub D, F	D -= F;	Subtrai	
imul D, F	D *= F;	Multiplica	
not D	D = ~D;	Inverte bits	
or D, F	D  = F;	Ou bit a bit	
and D, F	D &= F;	E bit a bit	
xor D, F	D ^= F;	Ou exclusivo bit a bit	
sal D, C	D <<= C;	Shift para a esquerda	shl
sar D, C	D >>= C; (D signed)	Shift para a direita com sinal	
shr D, C	D >>= C; (D unsigned)	Shift para a direita sem sinal	
cmp D, F	D - F; (só actualiza flags)	Compara valores	
test D, F	D & F; (só actualiza flags)	Compara bit a bit	
sete D8	D8 = zf;	Coloca 1 se igual, senão 0	setz
setne D8	D8 = ~zf;	Coloca 1 se diferente, senão 0	setnz
sets D8	D8 = sf;	Coloca 1 se negativo, senão 0	
setns D8	D8 = ~sf;	Coloca 1 se positivo, senão 0	
setg D8	D8 = ~(sf^of) & ~zf;	Coloca 1 se maior, senão 0	setnle
setge D8	D8 = ~(sf^of);	Coloca 1 se maior ou igual, ...	setnl
setl D8	D8 = (sf^of);	Coloca 1 se menor, senão 0	setnge
setle D8	D8 = (sf^of)   zf;	Coloca 1 se menor ou igual, ...	setng
seta D8	D8 = ~cf & ~zf;	Coloca 1 se superior, senão 0	setnbe
setae D8	D8 = ~cf;	Coloca 1 se superior ou igual, ...	setnb
setb D8	D8 = cf;	Coloca 1 se inferior, senão 0	setnae
setbe D8	D8 = cf   zf;	Coloca 1 se inferior ou igual, ...	setna
jmp F	eip = F;	Salta incondicionalmente	
je L	if(zf) eip = L;	Salta se igual (zero)	jz
jne L	if(~zf) eip = L;	Salta se diferente (não zero)	jnz
js L	if(sf) eip = L;	Salta se negativo	
jns L	if(~sf) eip = L;	Salta se positivo	
jg L	if(~(sf^of) & ~zf) eip = L;	Salta se maior	jnle
jge L	if(~(sf^of)) eip = L;	Salta se maior ou igual	jnl
jl L	if(sf^of) eip = L;	Salta se menor	jnge
jle L	if((sf^of)   zf) eip = L;	Salta se menor ou igual	jng
ja L	if(~cf & ~zf) eip = L;	Salta se superior	jnbe
jae L	if(~cf) eip = L;	Salta se superior ou igual	jnb
jb L	if(cf) eip = L;	Salta se inferior	jnae
jbe L	if(cf   zf) eip = L;	Salta se inferior ou igual	jna
call F	esp-=4; *esp=eip; eip=F;	Chama incondicionalmente	
ret	eip=*esp; esp+=4;	Retorna ao chamador	
leave	esp=ebp; ebp=*esp; esp+=4;	mov esp,ebp + pop ebp	

Dr – Destino (registo)

F – Fonte (imediato, registo ou memória)

D – Destino (registo ou memória)

Fm – Fonte (memória)

C – Contador (registo cl ou imediato)

Fv – Fonte (registo ou memória)

L – Local/Label em memória (imediato)

D8 – Destino a 8 bits (registo ou memória)

cf,zf,sf e of - Flags carry, zero, signal e overflow

Com dois operandos D e F não podem ser ambos de memória