Instruction	Opérande 1	Opérande 2	Effet
<code>mov</code>	<i>dst</i>	<i>src</i>	recopie <i>src</i> dans <i>dst</i>
<code>push</code>	<i>src</i>		place <i>src</i> au sommet de la pile
<code>pop</code>	<i>dst</i>		supprime le sommet de la pile et le place dans <i>dst</i>
<code>add</code>	<i>dst</i>	<i>src</i>	ajoute <i>src</i> à <i>dst</i>
<code>sub</code>	<i>dst</i>	<i>src</i>	soustrait <i>src</i> à <i>dst</i>
<code>mul</code>	<i>src</i>		multiplie <code>eax</code> par <i>src</i> (résultat dans <code>edx eax</code>)
<code>imul</code>	<i>src</i>		multiplie <code>eax</code> par <i>src</i> (résultat signé dans <code>edx eax</code>)
<code>imul</code>	<i>dst</i>	<i>src</i>	multiplie <i>dst</i> par <i>src</i> (signé)
<code>div</code>	<i>src</i>		divise <code>edx eax</code> par <i>src</i> (<code>eax</code> =quotient, <code>edx</code> =reste)
<code>idiv</code>	<i>src</i>		divise <code>edx eax</code> par <i>src</i> (complément à deux)
<code>inc</code>	<i>dst</i>		place <i>dst</i> + 1 dans <i>dst</i>
<code>dec</code>	<i>dst</i>		place <i>dst</i> - 1 dans <i>dst</i>
<code>neg</code>	<i>dst</i>		place - <i>dst</i> dans <i>dst</i>
<code>not</code>	<i>dst</i>		place (not <i>dst</i>) dans <i>dst</i>
<code>and</code>	<i>dst</i>	<i>src</i>	place (<i>src</i> AND <i>dst</i>) dans <i>dst</i>
<code>or</code>	<i>dst</i>	<i>src</i>	place (<i>src</i> OR <i>dst</i>) dans <i>dst</i>
<code>xor</code>	<i>dst</i>	<i>src</i>	place (<i>src</i> XOR <i>dst</i>) dans <i>dst</i>
<code>sal</code>	<i>dst</i>	<i>nb</i>	décalage arithmétique à gauche de <i>nb</i> bits de <i>dst</i>
<code>sar</code>	<i>dst</i>	<i>nb</i>	décalage arithmétique à droite de <i>nb</i> bits de <i>dst</i>
<code>shl</code>	<i>dst</i>	<i>nb</i>	décalage logique à gauche de <i>nb</i> bits de <i>dst</i>
<code>shr</code>	<i>dst</i>	<i>nb</i>	décalage logique à droite de <i>nb</i> bits de <i>dst</i>
<code>rol</code>	<i>dst</i>	<i>nb</i>	rotation à gauche de <i>nb</i> bits de <i>dst</i>
<code>ror</code>	<i>dst</i>	<i>nb</i>	rotation à droite de <i>nb</i> bits de <i>dst</i>
<code>cmp</code>	<i>sr1</i>	<i>sr2</i>	compare <i>sr1</i> et <i>sr2</i>
<code>jmp</code>	<i>adr</i>		saute vers l'adresse <i>adr</i>
<code>jxx</code>	<i>adr</i>		saute conditionné par <i>xx</i> vers l'adresse <i>adr</i>
<code>loop</code>	<i>adr</i>		répétition de la boucle <i>nb</i> de fois (<i>nb</i> dans <code>ecx</code>)

TABLE 1 – Les principales instructions en assembleur.

Signé				Non signé			
<code>je</code>	saute si vleft	=	vright	<code>je</code>	saute si vleft	=	vright
<code>jne</code>	saute si vleft	≠	vright	<code>jne</code>	saute si vleft	≠	vright
<code>j1, jnge</code>	saute si vleft	<	vright	<code>jb, jnae</code>	saute si vleft	<	vright
<code>jle, jng</code>	saute si vleft	≤	vright	<code>jbe, jna</code>	saute si vleft	≤	vright
<code>jg, jnle</code>	saute si vleft	>	vright	<code>ja, jnbe</code>	saute si vleft	>	vright
<code>jge, jnl</code>	saute si vleft	≥	vright	<code>jae, jnb</code>	saute si vleft	≥	vright

TABLE 2 – Sauts conditionnels usuels sur l'instruction `cmp` vleft, vright.