

Notice Utilisateur

Compiler le programme :

Entrer dans le terminal et se diriger dans le dossier stockant le programme
entrer la commande suivante pour compiler le projet si nécessaire:

gcc simu.c -o simulateur

puis :

./simulateur fichier.txt

où fichier.txt est le fichier contenant le code assembleur que vous voulez traduire
et exécuter

⚠ Les lignes du document fournis sont limitées à 100 caractères, au-delà des
erreurs seront détectées. Les nombres autorisés sont dans la plage
[-32 768;32 767].

Les instructions autorisées sont :

0	add rd, rn, S	$rd \leftarrow rn + S$
1	sub rd, rn, S	$rd \leftarrow rn - S$
2	mul rd, rn, S	$rd \leftarrow rn \times S$ (on ne prend que les 8 bits de poids faible de rn et S)
3	div rd, rn, S	$rd \leftarrow rn / S$ (division entière)
4	and rd, rn, S	$rd \leftarrow rn \& S$
5	or rd, rn, S	$rd \leftarrow rn S$
6	xor rd, rn, S	$rd \leftarrow rn \wedge S$
7	shl rd, rn, S	$rd \leftarrow rn$ décalé S fois à gauche (ou à droite si S est négatif), on remplace les nouveaux bits par 0.

10	ldb rd, (rn)S	$rd \leftarrow$ contenu de l'adresse (rn+S), sur 1 octet
11	ldw rd, (rn)S	$rd \leftarrow$ contenu de l'adresse (rn+S), sur 2 octets
12	stb (rd)S, rn	l'adresse (rd+S) reçoit le contenu de rn, sur 1 octet
13	stw (rd)S, rn	l'adresse (rd+S) reçoit le contenu de rn, sur 2 octets

20	jmp S	$PC \leftarrow S$
21	jzs S	$PC \leftarrow S$, si $Z=1$ (<i>Jump if Z Set</i>)
22	jzc S	$PC \leftarrow S$, si $Z=0$ (<i>Jump if Z Clear</i>)
23	jcs S	$PC \leftarrow S$, si $C=1$ (<i>Jump if C Set</i>)
24	jcc S	$PC \leftarrow S$, si $C=0$ (<i>Jump if C Clear</i>)
25	jns S	$PC \leftarrow S$, si $N=1$ (<i>Jump if N Set</i>)
26	jnc S	$PC \leftarrow S$, si $N=0$ (<i>Jump if N Clear</i>)

27	in rd	met dans rd une valeur entrée par l'utilisateur au clavier
28	out rd	affiche à l'écran en décimal le contenu du registre rd

29	rnd rd, rn, S	met dans rd un nombre aléatoire entier entre rn et S-1 inclus
31	hlt	termine l'exécution du programme