

TP compilation : ASM

Un programme assembleur pour la machine précédente est un fichier texte, donc chaque ligne répond à l'une de ces descriptions :

- Un identifiant suivi du symbole : (deux-points). Une telle ligne définit une étiquette symbolique qui permet de désigner l'adresse de l'instruction ou de la donnée située juste après l'étiquette. On désigne par *identifiant* une suite formée par des lettres, des chiffres et le symbole _ (souligné).
- Une ligne quelconque précédée du symbole #, qui est un commentaire.
- Un nombre entier positif, définissant la valeur initiale d'une donnée statique.
- Un nom d'instruction pris dans la colonne « nom » de la table présentée figure ??, suivi par ses paramètres séparés par des espaces. Sachant que :
 - un registre est désigné par les caractères \$r immédiatement suivis du numéro du registre,
 - une valeur immédiate est donnée soit directement par le nombre correspondant (deuxième paramètre de CONST), soit par une étiquette (deuxième paramètre de DIRECTREAD, premier paramètre de DIRECTWRITE).Exception pour l'instruction JUMPWHEN qui s'écrit JUMP \$rXX WHEN \$rYY (en remplaçant XX et YY par les numéros de registres correspondants).
- La nouvelle instruction ADDRESS, prenant en paramètres un registre r_d et une étiquette l , et qui a pour effet de placer dans le r_d l'adresse désignée par l'étiquette l .

Par convention les instructions du programme sont données en premier, s'achèvent avec une instruction EXIT, et sont suivies par les données. Chaque élément peut être précédé d'un nombre arbitraire d'espaces. La traduction d'un programme assembleur vers un programme en bytecode est exécutée par l'assembleur ./Assembler.

La figure 1 donne un exemple de programme.

```

# Borne et compteur
CONST $r0 5
CONST $r1 0
# Saut à l'étiquette 'test'
ADDRESS $r2 test
JUMP $r2
loop:
# Affichage de la valeur de 'a'
DIRECTREAD $r3 a
PRINT $r3
# Incrément et sauvegarde de 'a'
INCR $r3 2
DIRECTWRITE a $r3
# Incrément du compteur
INCR $r1 1
test:
# Test de non-dépassement de la borne
LT $r3 $r1 $r0
# Saut à l'étiquette 'loop' si non-dépassement
ADDRESS $r2 loop
JUMP $r2 WHEN $r3
# Affichage retour à la ligne
CONST $r0 10
PRINT $r0
# Fin
EXIT
# Une donnée allouée statiquement
a:
97

```

FIGURE 1 – Exemple de programme ASM