

Activité – Architecture de Von Neumann

Exercice 1

a. Nombre d'octets pour chaque objet :

- Livre : $300 \text{ pages} \times 1500 \text{ caractères/page} \times 1 \text{ octet/caractère} = 4,5 \cdot 10^5 \text{ o}$
- Photo : $4000 \times 3000 \text{ pixels} \times 4 \text{ octets/pixel} = 4,8 \cdot 10^7 \text{ o}$
- Vidéo : $5 \text{ minutes} \times 20 \text{ Méga-o/minute} = 100 \text{ Méga-o} = 10^8 \text{ o}$
- Film : $120 \text{ min} \times 20 \text{ Mo/minute} = 2400 \text{ Mo} = 2,4 \text{ Go} = 2,4 \cdot 10^9 \text{ o}$

b. Il suffit de diviser la taille de la mémoire par la taille de l'objet (en division euclidienne) :

Livre : $500 \cdot 10^9 / 4,5 \cdot 10^5 = 1,1 \cdot 10^6 \text{ livres}$

Vidéo : $500 \cdot 10^9 / 10^8 = 5000 \text{ vidéos}$

Photo : $500 \cdot 10^9 / 4,8 \cdot 10^7 = 10\,417 \text{ photos}$

Film : $500 \cdot 10^9 / 2,4 \cdot 10^9 = 208 \text{ films}$

c. Conversion en puissance de 2 :

- Livre : $2^{18} < 4,5 \cdot 10^5 \text{ o} < 2^{19}$, donc

$$4,5 \cdot 10^5 \text{ o} = \frac{4,5 \cdot 10^5}{2^{18}} \times 2^{18} = \underline{1,72 \times 2^{18} \text{ o}} = 1,72 \times 2^8 \times 2^{10} = 1,72 \times 256 \times 2^{10} = \underline{439,4 \text{ ko}}$$

- Photo : $2^{25} < 4,8 \cdot 10^7 \text{ o} < 2^{26}$, donc

$$4,8 \cdot 10^7 \text{ o} = \frac{4,8 \cdot 10^7}{2^{25}} \times 2^{25} = \underline{1,43 \times 2^{25} \text{ o}} = 1,43 \times 2^5 \times 2^{20} = 1,43 \times 32 \times 2^{20} = \underline{45,8 \text{ Mo}}$$

- Vidéo : $2^{26} < 10^8 \text{ o} < 2^{27}$, donc

$$10^8 \text{ o} = \frac{10^8}{2^{26}} \times 2^{26} = \underline{1,49 \times 2^{26} \text{ o}} = 1,49 \times 2^6 \times 2^{20} = 1,49 \times 64 \times 2^{20} = \underline{95,4 \text{ Mo}}$$

- Film : $2^{31} < 2,4 \cdot 10^9 \text{ o} < 2^{32}$, donc

$$2,4 \cdot 10^9 \text{ o} = \frac{2,4 \cdot 10^9}{2^{31}} \times 2^{31} = \underline{1,12 \times 2^{31} \text{ o}} = 1,12 \times 2^1 \times 2^{30} = \underline{2,2 \text{ Go}}$$

Activité – Circuits combinatoires

Exercice 2

a.

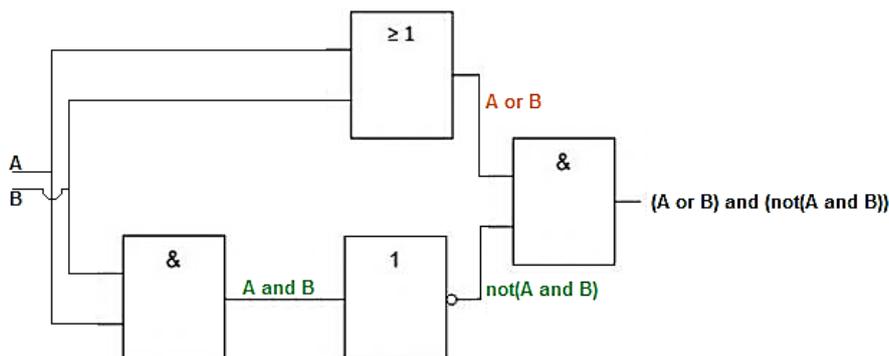
A	B	A xor B
False	True	True
False	False	False
True	True	False
True	False	True

b.

A	B	A or B	A and B	not(A and B)	(A or B) and (not(A and B))
False	True	True	False	True	True
False	False	False	False	True	False
True	True	True	True	False	False
True	False	True	False	True	True

On a donc bien $A \text{ xor } B = (A \text{ or } B) \text{ and } (\text{not}(A \text{ and } B))$

c. Le circuit est alors :



TP – Introduction à l'assembleur

Exercice 3

```
LDR R0, 50
LDR R1, 51
CMP R0, R1
JMPN Neg
MOV R2, R0
JMP Suite
```

Neg:

```
MOV R2, R1
JMP Suite
```

Suite:

```
STR R2, 52
HALT
```

Mettre la valeur x dans R0

Mettre la valeur y dans R1

Comparer la valeur de R0 à la valeur de R1

Si R0 < R1 aller au label Neg

Mettre la valeur de R0 dans R2

Aller au label Suite

Label Neg :

Mettre la valeur de R1 dans R2

Aller au label Suite

Label Suite :

Transférer la valeur contenue dans R2 à la case mémoire 52

Terminer le programme

Ce programme stocke la plus grande des deux valeurs x et y dans la case mémoire 52.

Exercice 4

a.

```
LDR R0, 50
LDR R1, 51
JMP A
```

A:

```
SUB R2, R0, R1
CMP R2, #0
JMPP A
JMPN B
STR #0, 52
```

B:

```
ADD R2, R2, R1
STR R2, 52
HALT
```

Mettre la valeur x dans R0

Mettre la valeur y dans R1

Aller au label A

Label A :

Soustraire R1 à R0 et stocker le résultat dans R2

Regarder si le résultat est positif

Si R2 > 0, aller au label A

Si R2 < 0, aller au label B

Dans les autres cas (R2 = 0), transférer 0 à la case mémoire 52

Label B :

Additionner R1 à R2 et stocker le résultat dans R2

Transférer la valeur contenue dans R2 à la case mémoire 52

Terminer le programme

Ce programme calcule le reste de la division euclidienne de x par y et stocke le résultat dans la case mémoire 52.

b. Pour avoir le quotient de la division euclidienne de x par y, il faut compter le nombre de fois où le résultat de x - y est positif. Par exemple :

```
LDR R0, 50
LDR R1, 51
LDR R3, #0
JMP A
```

A:

```
SUB R2, R0, R1
CMP R2, #0
ADD R3, R3, #1
JMPP A
JMPN B
STR #0, 52
STR R3, 53
```

B:

```
ADD R2, R2, R1
SUB R3, R3, #1
STR R2, 52
STR R3, 53
HALT
```

Mettre la valeur x dans R0

Mettre la valeur y dans R1

Mettre la valeur 0 dans R3 (compteur)

Aller au label A

Label A :

Soustraire R1 à R0 et stocker le résultat dans R2

Regarder si le résultat est positif

Ajouter 1 au compteur R3

Si R2 > 0, aller au label A

Si R2 < 0, aller au label B

Dans les autres cas (R2 = 0), transférer 0 à la case mémoire 52

Transférer la valeur contenue dans R3 à la case mémoire 53

Label B :

Additionner R1 à R2 et stocker le résultat dans R2

Soustraire 1 au compteur R3

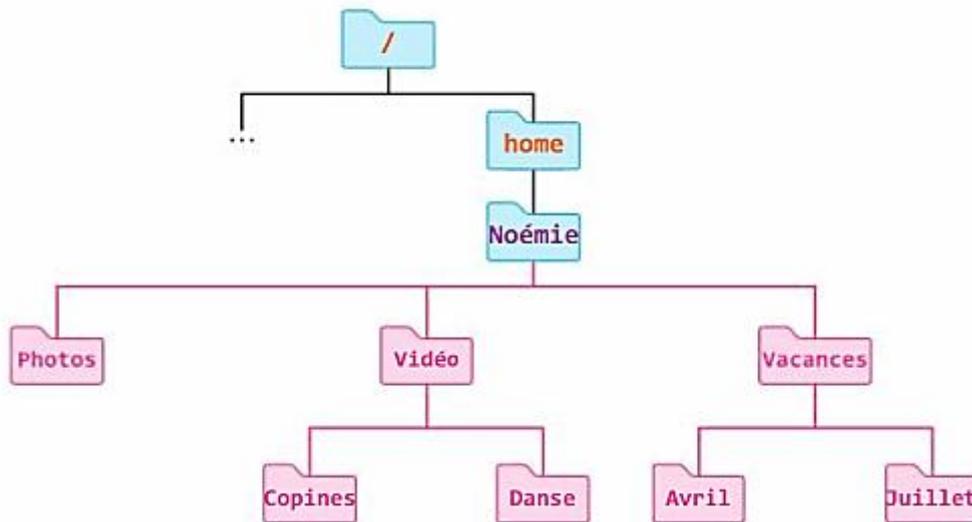
Transférer la valeur contenue dans R2 à la case mémoire 52

Transférer la valeur contenue dans R3 à la case mémoire 53

Terminer le programme

Exercice 5

<code>cd /home/Noémie</code>	Aller dans le dossier Noémie
<code>mkdir Photos</code>	Créer un dossier Photos dans le dossier Noémie
<code>mkdir Vidéo</code>	Créer un dossier Vidéo dans le dossier Noémie
<code>cd Vidéo</code>	Aller dans le dossier Vidéo
<code>mkdir Copines</code>	Créer un dossier Copines dans le dossier Vidéo
<code>cd ..</code>	Revenir dans le dossier Noémie
<code>mkdir Vacances</code>	Créer un dossier Vacances dans le dossier Noémie
<code>mkdir Vacances/Avril</code>	Créer un dossier Avril dans le dossier Vacances
<code>mkdir Vacances/Juillet</code>	Créer un dossier Juillet dans le dossier Vacances
<code>mkdir /home/Noémie/Vidéo/Danse</code>	Créer un dossier Danse dans le dossier Vidéo



Exercice 6

On considère l'arborescence ci-contre :

- a. Le résultat est f2 e.
- b. Il faut aller dans le répertoire *g* avec `cd ../c/g`
- c. Il faut exécuter `ls b`
- d. Il faut exécuter `cp ../f1 ../a/d`
- e. Il faut aller dans le répertoire *f* avec `cd ../c/f`
- f. Il faut d'abord supprimer les fichiers du répertoire *d* avec :
`cd a/d`
`rm *`
 Puis on supprime le répertoire *d* avec :
`cd ..`
`rmdir d`
 Puis on supprime le répertoire *a* avec :
`cd ..`
`rmdir a`
- g. Il faut exécuter `mv c/g/* b/e`