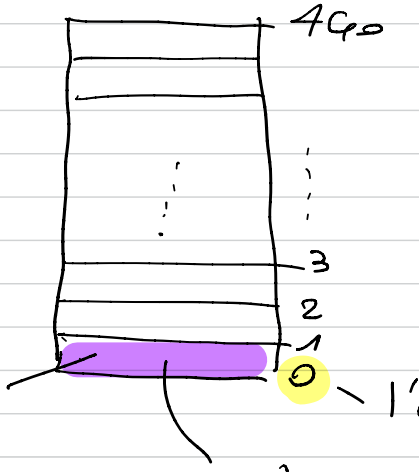
INFO1 - cours - SCANF en détail 13.X.2020

Syntaxe : ret = scanf ("format", &variable);

Mémoire 4Go RAM

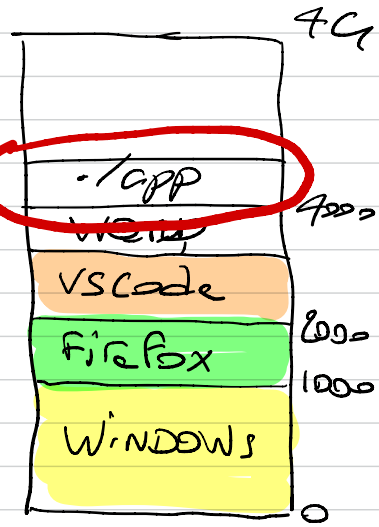


4 * 1024 * 1024 * 1024

emplacement mémoire

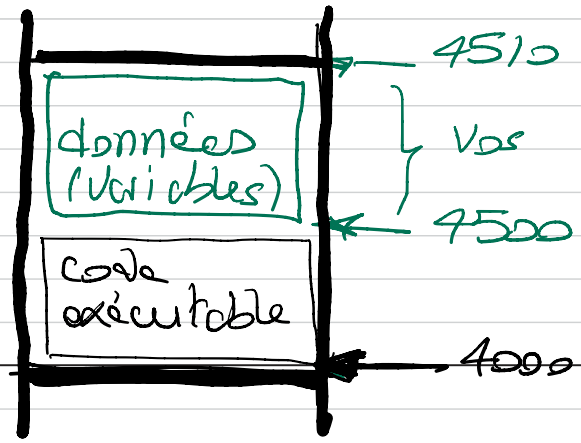
l'adresse à cet emplacement

contenu de l'emplacement



programme pas fait pour s'exécuter n'importe où en mémoire

Programme Relogeable



vos variables du programme

```
uint16_t x = 0;
printf("x= %hu \n", x);        //x=0
printf("adress of x: %p \n", &x);
```

adresse de la variable x

TD 2020/013

15h13

```
x = 0
adress of x = 0x7ffcc16779de
```

base 16 (hexadecimal)

ajouter sous la déclaration de x :

3 autres variables de type uint16_t

y = 0
z = 0
t = 0

afficher les 4 adresses de x, y, z et t.

15h23

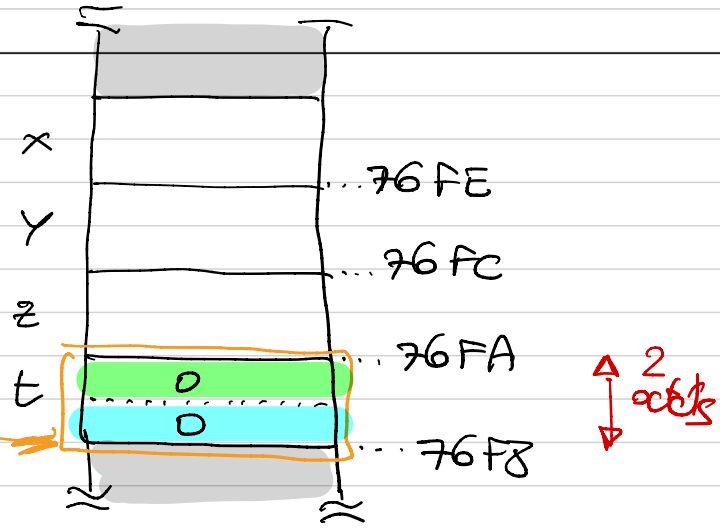
```
adress of x = 0x7ffe6d3d76fe
adress of y = 0x7ffe6d3d76fc
adress of z = 0x7ffe6d3d76fa
adress of t = 0x7ffe6d3d76f8
```

2 octets
2 octets
2 octets

uint16_t → 16 bit → 2 octets / variable.

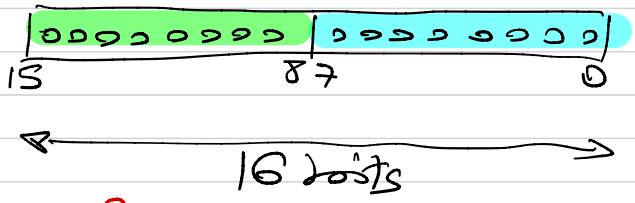
uint8_t : %hhu	int8_t : %hhd
uint16_t : %hu	int16_t : %hd
uint32_t : %u	int32_t : %d
uint64_t : %lu	int64_t : %ld

```
adress of x = 0x7ffe6d3d76fe
adress of y = 0x7ffe6d3d76fc
adress of z = 0x7ffe6d3d76fa
adress of t = 0x7ffe6d3d76f8
```




```
uint16_t t = 0;
```

```
uint16_t t = 0;
```



valeur initiale

```
int32_t d = 8;
```

identificateur

type

&d adresse de d en mémoire

Pourquoi le scanf a-t-il besoin de l'adresse de la variable ?

1. Récupère l'entrée clavier
2. Décodage selon le format indiqué
3. Mettre à jour la variable

adresse de la boîte

il faut donner au scanf les moyens d'atteindre la variable en mémoire

→ lui donner son adresse

printf(" Entrez z : ");
scanf("%hd", &z);

→ affiche x, y, z et t.

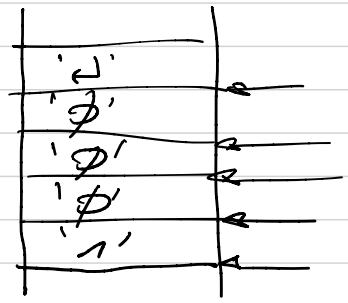
15h42.

```
printf("Enter z: ");
ret = scanf("%hd", &z);
printf("x = %hu\n", x);
printf("y = %hu\n", y);
printf("z = %hu\n", z);
printf("t = %hu\n", t);
```

← valeur initiale

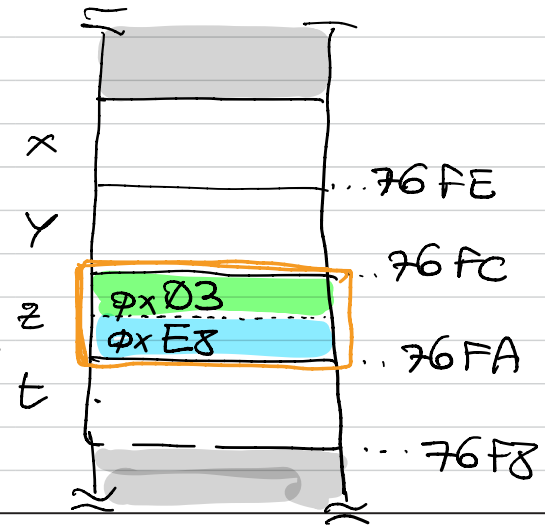
Enter z : 1000 ↵

buffer clavier



$\boxed{1000}_{10} \rightarrow 03E8_{16}$

%hu → 2 octets

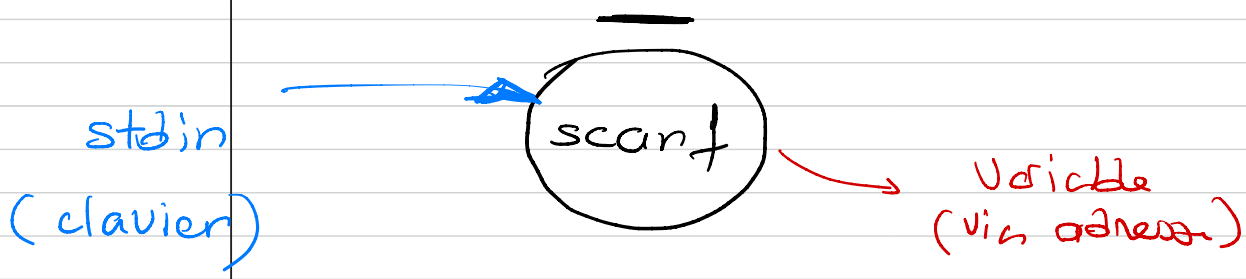


Erreur volontaire : ~~%hu~~ → %u

~~1000~~ → 123456

ret = scanf("format", &variable);

1. Connaître exactement le type de la variable
2. Choisir le format en correspondance



ex : "F16" ("F%d" &x);

x12y34 ("x%d y%d" &x &y);

./app 120 100

argv[1] argv[2]

largeur = atof(argv[1])

"120"
↓
120.0

Pas de contrôle possible

./app toto 100

Pas convertible avec atof

./app 120 100

argv[1] = "120"

chaine de caractères.

ret = sscanf(arg[1] , "%Pf" , &largeur);

string scanf

Tester si ret == 1

(II) créer une variable W de type double

récupérer le param argv[1] → W

tester si le param est correct ou non

Si pas correct → afficher "ERRORR"

si correct → afficher la valeur de W

16h22