

word → tableau dynamique

nb char : nb mot du texte

nb mot = $WC(\text{text})$; // 200

Sans doublons: 107 mots ≠ .

word: NULL condition initiale.

Pour chaque mot non vu: **realloc**

1^e mot: $\text{ptr} = \text{realloc}(\text{word}, \boxed{\text{dim}})$;

si $\text{ptr} \neq \text{NULL}$ → $\text{word} = \text{ptr}$

"remplir word".

dim

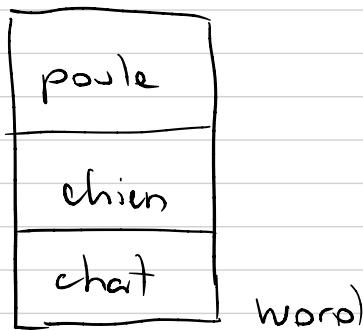
$\text{wordIndex} = \emptyset$

dim: $(\text{wordIndex} + 1) * \text{sizeof}(\text{char})$

L66: $\text{word} = \text{malloc} \rightarrow \text{word} = \text{NULL}$.

L118: méconnaissance de realloc -
 + test ptr non null.

15^h
17

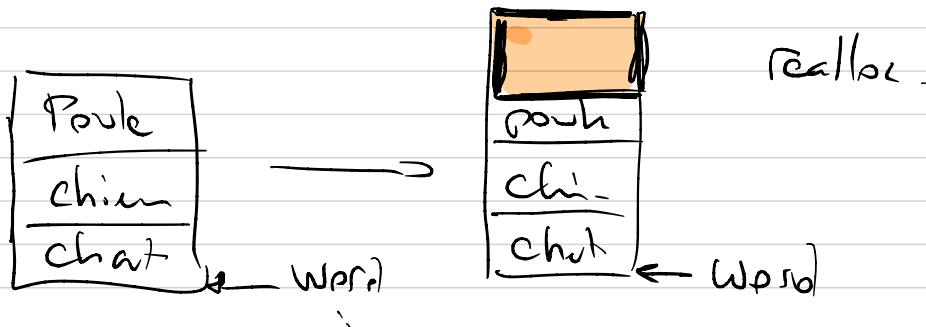


1) nonseen mot = chien.

bool wordAlreadyExist = TRUE \Rightarrow on passe au mot suivant

2) nonseen mot = lapin

word Already Exist = FALSE \rightarrow il faut ajouter ce mot dans Word.



wordIndex = 3

wordIndex = 4

$\text{ptr} = (\text{char} *) \text{realloc}(\text{word}, (\text{wordIndex} + 1) * \text{sizeof}(\text{char} *))$;
si $\text{ptr} \neq \text{NULL} \Rightarrow \text{word} = \text{ptr}$

Bilan des zones allouées dynamiquement.

[main]

- ① zone de texte lire depuis le fichier

[ws]

- ② tableau word (realloc)

[ws]

- ③ word[k] mots uniques

Liberation

① zone à texte : à la fin. ③

② word : en sortie de [ws] après avoir libérer les Word[k]

③ word[k] ①

→ boucle pour tous les éléments
de word. (for)

```
if (ptr != NULL) {  
    free (ptr);  
    ptr = NULL;  
}
```

15'45

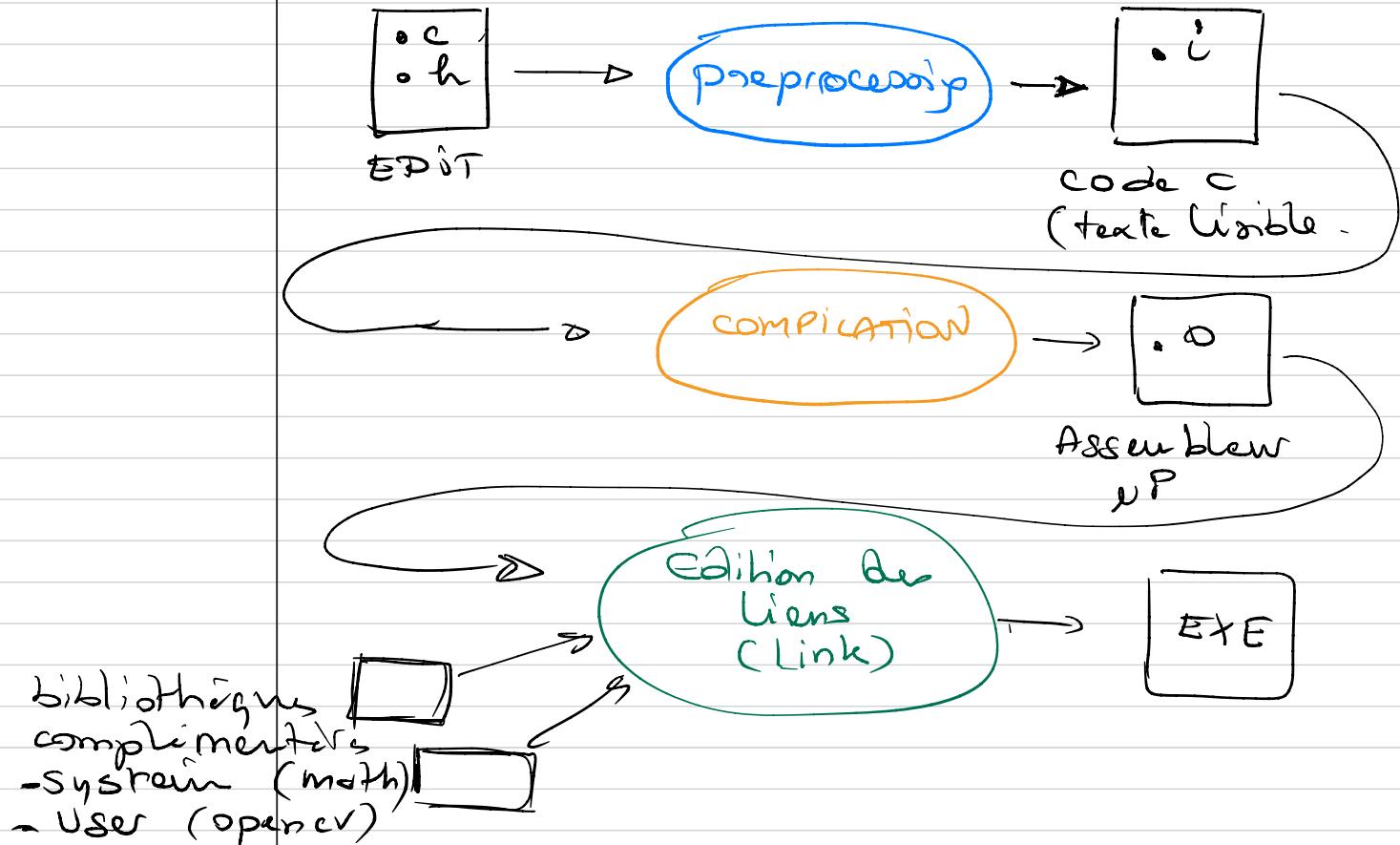
Preprocessor

include < > fichier .h système
fichier .h user

Jamais include de .c

define N 100 — texte de remplacement
' symbol du préprocesseur

Cycle de génération de l'exécutable



gcc -E test.c > test.i

preprocessing only

SYMBOL → ()

#define SYMBOL text replacement

#define PI (3.141592654)

TD 2020 0907

#define M (80+20) ← (h)
int32_t k = 3 * M] ← (c)
afficher la valeur de k.] 16^h10.
(260).

int32_t k = 3 * 80 + 20
240 + 20 = 260.

k = 3 * (80+20)
3 * 100 = 300.



#define SYMBOL (TEXT)

MACRO

```
#define SIGN(x) ((x) >= 0 ? (1) : (-1))
```

↑ Espace

$\text{SIGN}(x)$: +1 si $x \geq 0$
 -1 sinon.

```
double z = -9.1;
```

```
int s = SIGN(z);
```

Preprocessing

```
double s = ((z) >= 0 ? (1) : (-1));
```

```
scanf("%lf", &z);
```

```
s = SIGN(z);
```

```
scanf("%lf", &z);
```

```
s = ((z) >= 0 ? (1) : (-1));
```



! Pay attention
to type

```
char c = 30;
```

```
int s = SIGN(c);
```

Macro avec
plusieurs
param

```
#define SOMME(x,y,z) ((x)+(y)+(z))
```

```
int sign(double x) {
```

espace

```
    return x >= 0 ? 1 : -1;
```

}

Macro : MAX2 pow 2 parameters

m = MAX2 (48, 27);

Macro : MAX3 pow 3 parameters

m = MAX3 (48, 27, 300);

Add the 6 values of m.

?

16⁸¹

#define MAX2 (a,b) ((a) > (b)) ? (a) : (b)

int m = MAX2 (48, 27);
m = (48) >= (27) ? (48) : (27);

#define MAX3 (a,b,c) (MAX2 (MAX2 (a), b)), (c))

MAX3 (a,b,c) ((a) > (b)) ? (a) : ((b) > (c)) ? (b) : (c))