

Get up to speed with this
fun and friendly road map to the technology

Introduction au C

FOR
DUMMIES[®]

CD-ROM loaded
with goodies

**A Reference
for the
Rest of Us!**

B. Quoitin
Ecole Polytechnique de Louvain
Université catholique de Louvain



Table des matières

- Lecture de chaînes de caractères
- Compilation conditionnelle
- Fichiers headers (.h)
- Pointeurs de fonctions
- Enumérations, unions, bit fields
- La sexualité des ordinateurs (endianness)
- malloc() is not free
- Suggestion d'exercices

Lecture de chaînes de caractères

scanf()

- Rappel

```
char * ligne;
```

```
scanf("%s", ligne);
```



PLOUF

```
char * ligne= malloc(100);
```

```
scanf("%s", ligne);
```

```
free(ligne);
```

```
char ligne[100];
```

```
scanf("%s", ligne);
```

scanf()

- Version plus sûre

```
char * ligne= malloc(100);  
  
scanf("%99s", ligne);  
  
free(ligne);
```

Indiquer à scanf() qu'au maximum 99 caractères peuvent être lus (+ '\0' terminal)

```
char ligne[100];  
  
scanf("%99s", ligne);
```

scanf()

- Du bon usage de scanf()

```
char ligne[100];
int num;

printf("Entrez une chaîne: ");
num= scanf("%99s", ligne);
if (num < 1) {
    printf("Erreur: aucune chaîne entrée\n");
    exit(EXIT_FAILURE);
}
printf("La chaîne entrée est \"%s\"\n", ligne);
```

```
Entrez une chaîne: coco est content
La chaîne entrée est "coco"
```

fgets()

- Lecture d'une ligne entière

```
char line[100];

printf("Entrez une chaîne: ");
if (fgets(line, sizeof(line), stdin) != NULL) {

    printf("La chaîne entrée est \"%s\\n\"", ligne);

}
```

```
Entrez une chaîne: coco est content
La chaîne entrée est "coco est content
"
```

fgets()

- Pas si simple la lecture d'une ligne...

```
FILE * stream= stdin;
char line[100];
char ch;

if (fgets(line, sizeof(line), stream) != NULL) {

    /* Vérifie la présence de la fin de ligne. */
    if ((strchr(line, '\n') == NULL) && !feof(stream)) {

        /* Si pas trouvé, vider le flux (see STR35-C). */
        printf("Avertissement: la ligne est trop longue\n");
        while (((ch= fgetc(stream)) != '\n') &&
                !feof(stream) && !ferror(stream));

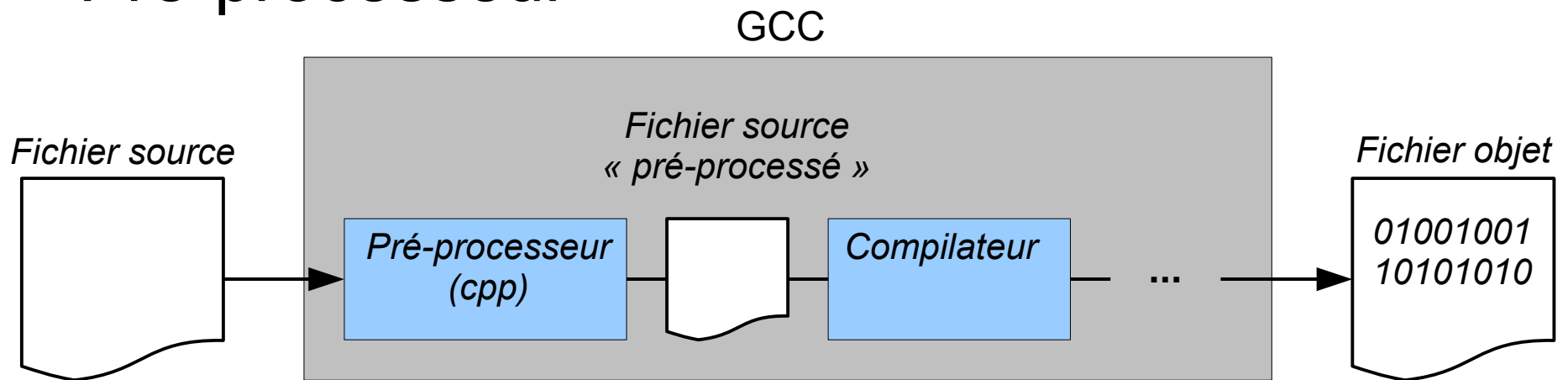
    }

}
```


Compilation conditionnelle

Compilation conditionnelle

- Pré-processeur



- Utiliser `#ifdef` ou `#ifndef`

- Générer différentes version d'un programme
p.ex. avec/sans infos de débogage ou utilisant
des fonctions spécifiques de la plateforme

Compilation conditionnelle

- Syntaxe

```
#ifdef / #ifndef SYMBOLE  
  
/* code */  
  
#endif
```

```
#ifdef / #ifndef SYMBOLE  
  
/* code */  
  
#else  
  
/* code */  
  
#endif
```

Définition de symboles

- Dans le code

```
#define SYMBOLE  
  
#define SYMBOLE valeur_du_symbole
```

- Exemples

```
#define DEBUG  
  
#define DEBUG_LEVEL 2
```

Définition de symboles

- En ligne de commande (via gcc)

```
gcc -DSYMBOLE ...
```

```
gcc -DSYMBOLE=valeur_du_symbole ...
```

- Exemples

```
gcc -DDEBUG ...
```

```
gcc -DDEBUG_LEVEL=2 ...
```

Définition de symboles

- Définis par le pré-processeur/compilateur

```
__FILE__  
__LINE__  
__DATE__  
__TIME__  
  
__linux__, __MACH__, __FreeBSD__, __CYGWIN__, __SVR4, ...  
  
__GNUC__  
  
__cplusplus  
  
...
```

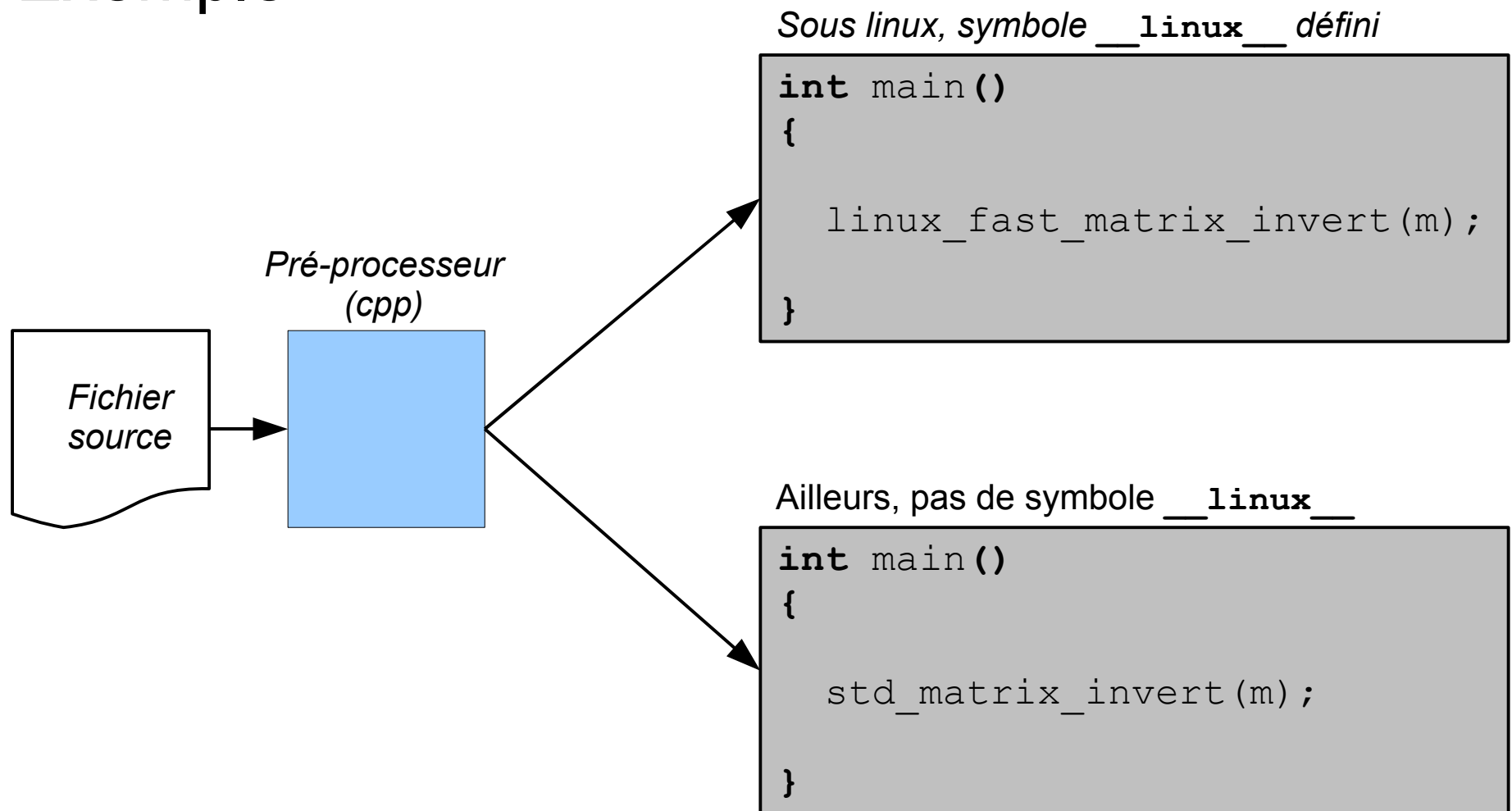
Compilation conditionnelle

- Exemple
 - Supposons l'existence sous linux d'une fonction d'inversion de matrice super-rapide
 - Détection de la présence de linux

```
int main()  
{  
  
#ifdef __linux__  
    linux_fast_matrix_invert(m);  
#else  
    std_matrix_invert(m);  
#endif /* __linux__ */  
  
}
```

Compilation conditionnelle

- Exemple



Compilation conditionnelle

- Exemple

```
int fct()  
{  
  
#ifdef DEBUG  
    printf("DEBUG: je passe à la ligne %d de %s\n",  
           __LINE__, __FILE__);  
#endif  
  
}
```

- Version debug: gcc -DDEBUG ...
- Version finale: gcc ...

Compilation conditionnelle

- Exemple

```
#ifdef DEBUG
#define pdebug(M) \
    printf("DEBUG: %s (ligne %d, %s)\n", \
           M, __LINE__, __FILE__);

#else
#define pdebug(M)
#endif
```

Fichiers Headers (.h)

Fichiers headers

- **Exemple:** `stdio.h`

```
#include <stdio.h>

int main()
{
    printf("The world is flat (T. Friedman)\n");
    return 0;
}
```

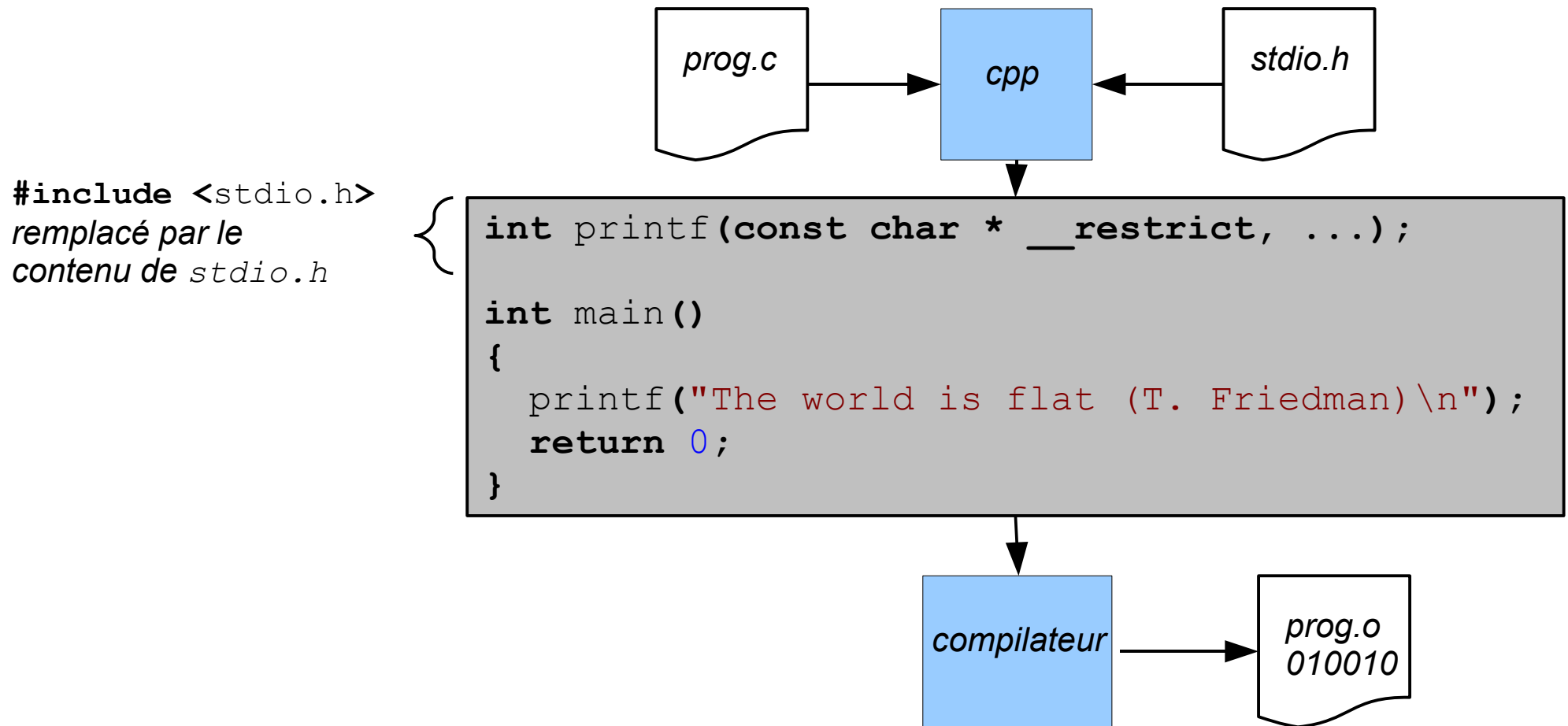
Note: Le fichier header `stdio.h` contient (entre autres) la signature (le prototype) de `printf()`

```
int printf(const char * __restrict, ...);
```

(voir `/usr/include/stdio.h`)

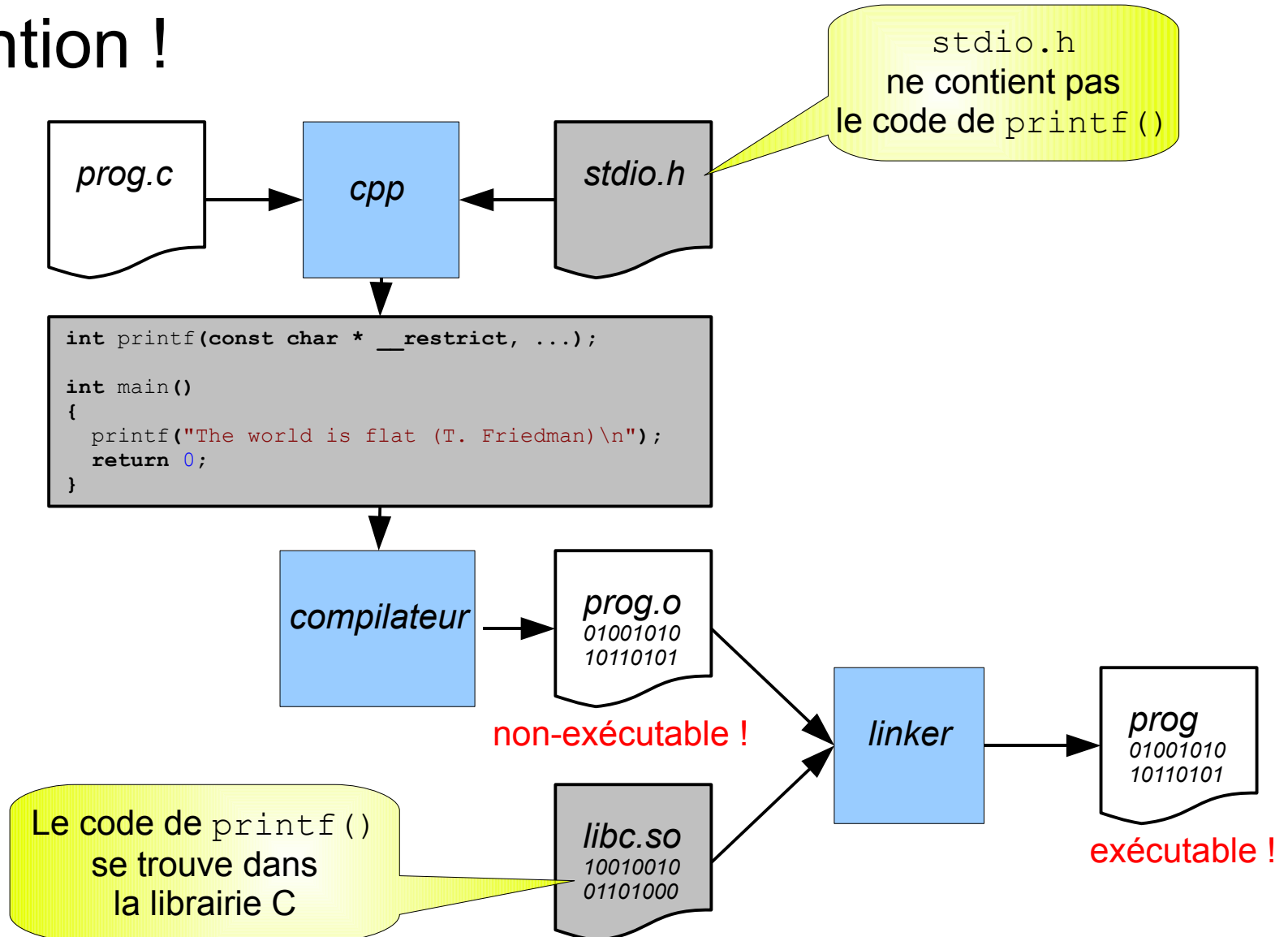
Fichiers headers

- Exemple: `stdio.h`



Fichiers headers

- Attention !



Fichiers headers

- Découpage d'un programme

util.c

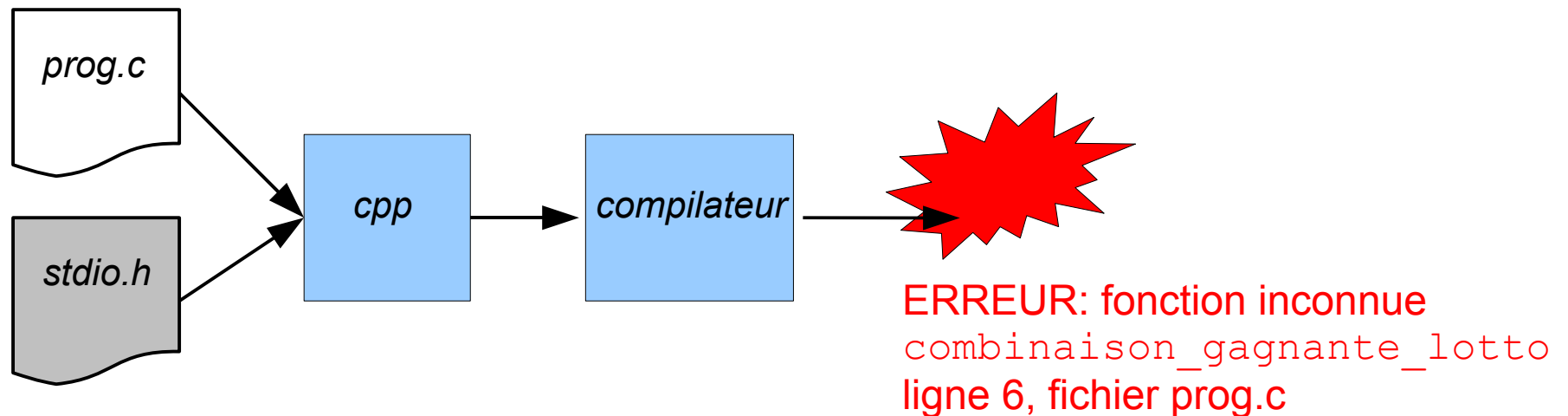
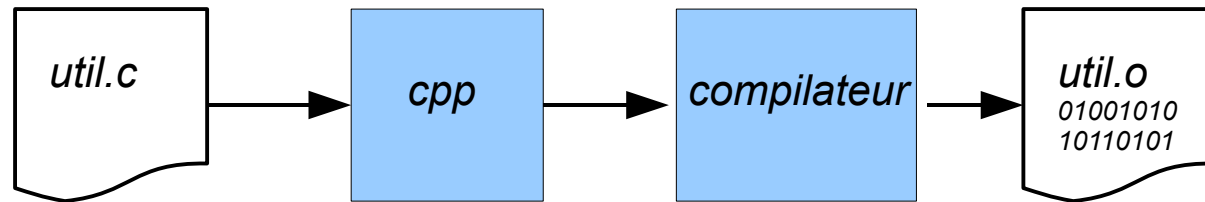
```
void combinaison_gagnante_lotto(int * tableau,  
                                size_t taille)  
{ /* confidentiel */ }
```

prog.c

```
#include <stdio.h>  
  
int main()  
{  
    int tableau[7], index;  
    combinaison_gagnante_lotto(tableau, 7);  
    printf("La combinaison gagnante est ");  
    for (index= 0; index < 7; index++)  
        printf(" %d", tableau[index]);  
    printf(" )\n");  
    return 0;  
}
```

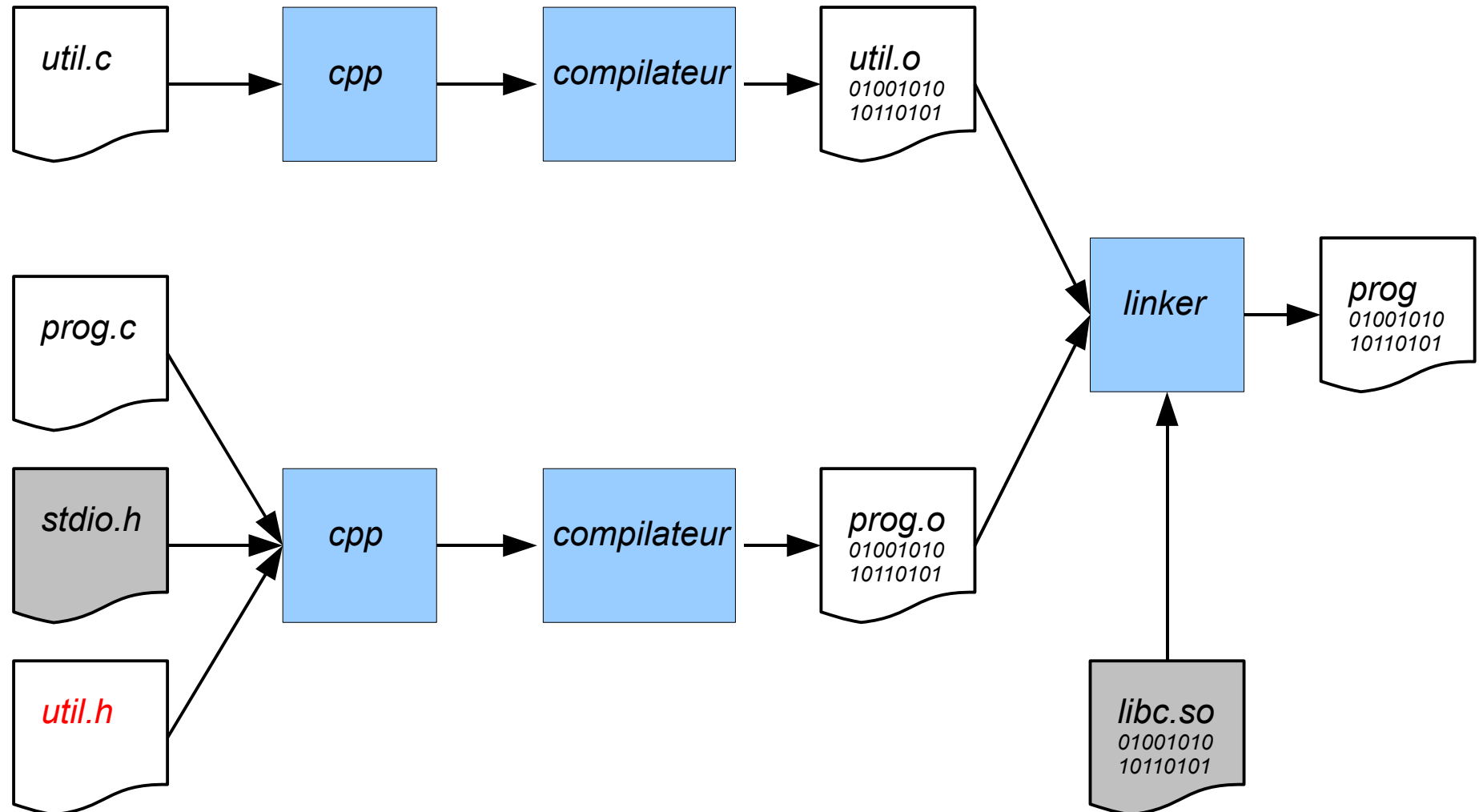
Fichiers headers

- Découpage d'un programme



Fichiers headers

- Découpage d'un programme



Fichiers headers

- Découpage d'un programme

util.h

```
#ifndef __UTIL_H__
#define __UTIL_H__

void combinaison_gagnante_lotto(int * tableau,
                                size_t taille);

#endif /* __UTIL_H__ */
```

Note: directives `#ifndef` et `#define` empêchent l'inclusion multiple du fichier header.

Pointeurs de fonctions

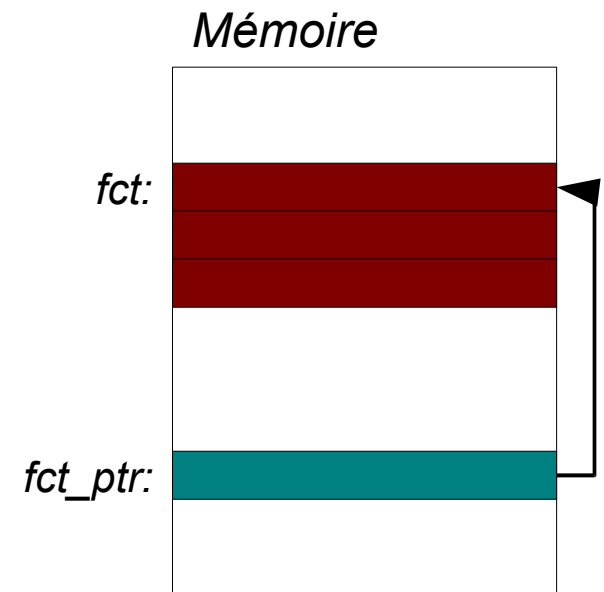
Pointeurs de fonctions

- Définition

```
type (* identifiant) ( [ arguments ] );  
  
typedef type (* identifiant_type) ( [ arguments ] );
```

- Exemple

```
void fct(char * str, char ** carptr) {  
    *carptr= str + strlen(str);  
}  
  
int main() {  
    void (*fct_ptr)(char *, char **) = fct;  
    char * plop;  
    fct_ptr("Quoi d'neuf docteur ?", &plop);  
    return 0;  
}
```



Pointeurs de fonctions

- Exemple

```
#include <signal.h>
#include <stdio.h>

int fini= 0;

void signal_handler(int signum) {
    fini= 1;
}

int main()
{
    signal(SIGINT, signal_handler);
    while (!fini) {}
    return 0;
}
```

Pointeurs de fonctions

- Exemple

```
typedef int (*compare)(int a, int b);

int compare_croissant(int a, int b) {
    if (a < b) return -1; else if (a > b) return 1;
    return 0;
}

int compare_decroissant(int a, int b) {
    if (a < b) return 1; else if (a > b) return -1;
    return 0;
}

int main() {
    int tableau[100];
    trie(tableau, 100, compare_decroissant);
    return 0;
}
```

Enumérations, unions et bit fields


Enumérations

- Définition

```
enum {  
    label (, label )*  
} identifiant ;  
  
typedef enum {  
    label (, label )*  
} identifiant_type ;
```

- Exemple

```
typedef enum {  
    ROUGE,  
    VERT,  
    BLEU  
} type_couleur;
```



ROUGE vaudra 0
VERT vaudra 1
BLEU vaudra 2

Enumérations

- Exemple

```
typedef enum {  
    PAS_D_ERREUR                = 0,  
  
    ERREUR_LECTURE_FICHER = 100,  
    ERREUR_ECRITURE_FICHER= 101,  
  
    ERREUR_SYNTAXE            = 200,  
  
    /* ... */  
} type_erreur;
```

Deux valeurs identiques
= pas permis !

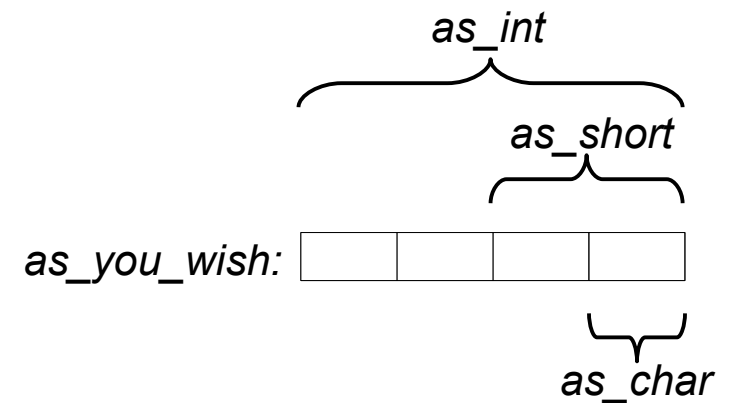
Unions

- Définition

```
union {  
    ( type identifiant ; )+  
} identifiant ;
```

- Exemple

```
union {  
    int    as_int;  
    char   as_char;  
    short  as_short;  
} as_you_wish ;
```



sizeof(as_you_wish)
vaut 4

Unions

- Exemple d'application

```
#define MODE_INT      1
#define MODE_STRING  2

typedef union {
    int    as_int;
    char * as_string;
} param_t;

void fct(int mode, param_t param) {
    switch (mode) {
        case MODE_INT:
            do_something_int(param.as_int);
            break;
        case MODE_STRING:
            do_something_str(param.as_string);
            break;
    }
}
```

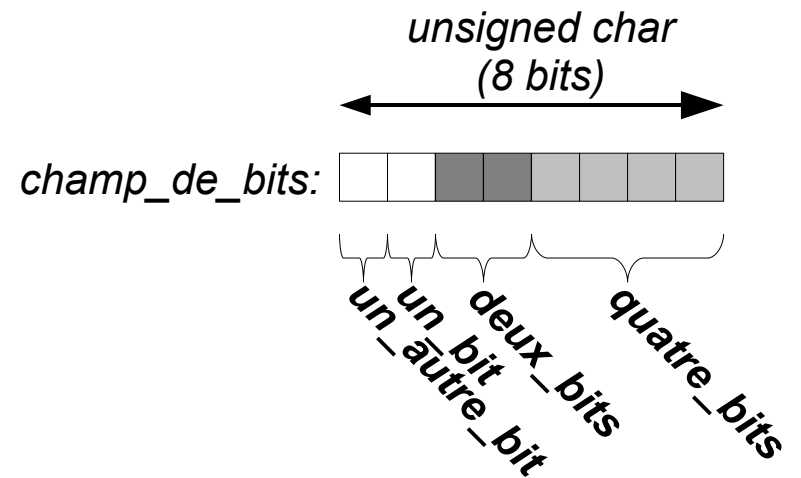
Bit fields

- Définition

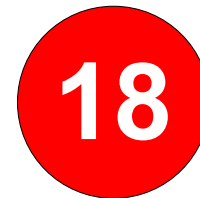
```
struct {  
    ( type identifiant : longueur ; )+  
} identifiant ;
```

- Exemple

```
struct {  
    unsigned char quatre_bits : 4;  
                    deux_bits  : 2;  
                    un_bit     : 1;  
                    un_autre_bit: 1;  
} champs_de_bits ;
```



La sexualité des ordinateurs (endianness)

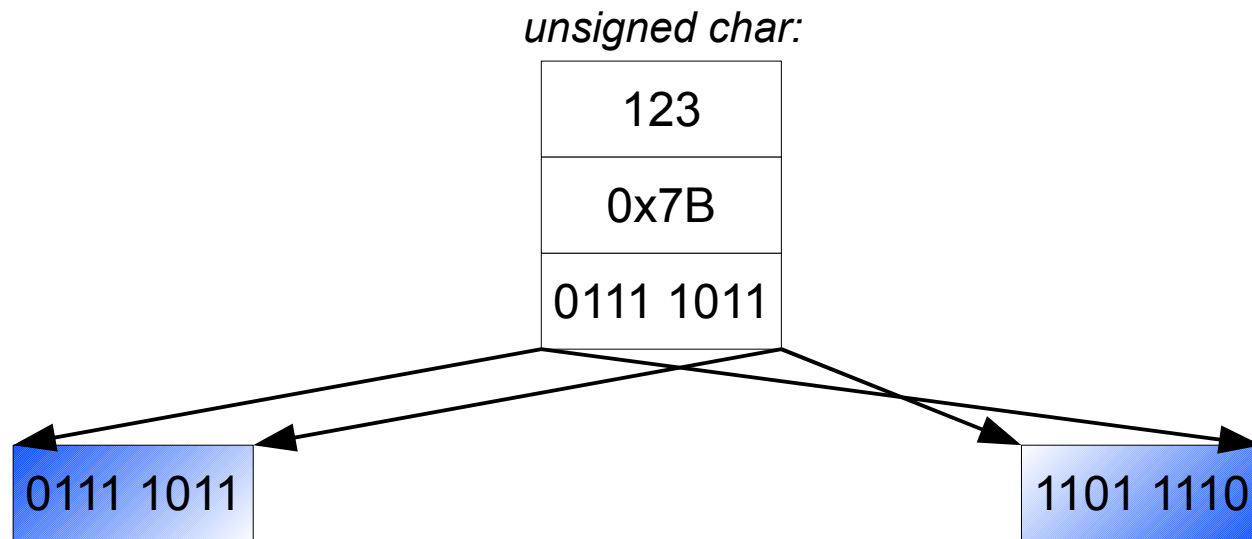


contenu déconseillé
aux moins de 18 ans

Endianness

Big-endian

Little-endian



Exemples:

little-endian: i386

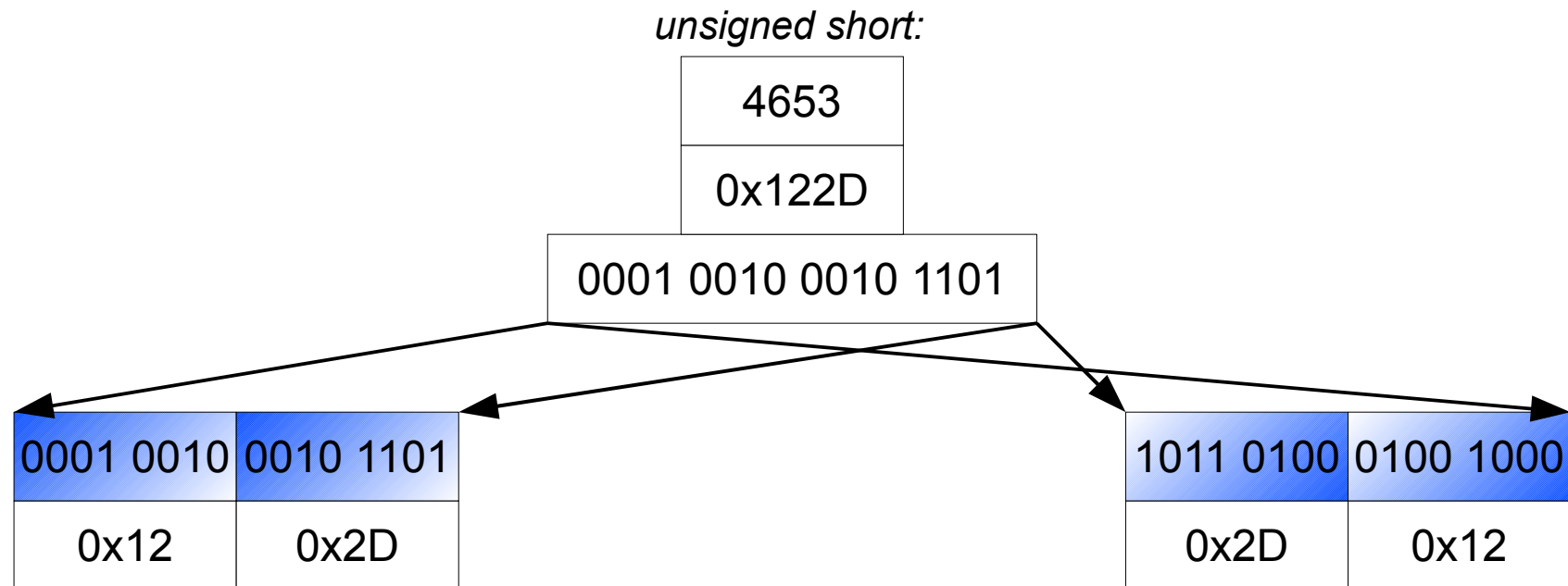
big-endian: SPARC, PowerPC

*-endian: ARM

Endianness

Big-endian

Little-endian



Endianness

```
#include <assert.h>
#include <fcntl.h>
#include <stdint.h>
#include <stdio.h>
#include <unistd.h>

int main() {
    uint32_t var= 0xFEBA8721;
#ifdef READER
    int fd= open("plop.bin", O_RDONLY, 0);
    assert(fd >= 0);
    assert(read(fd, &var, sizeof(var)) >= 0);
    printf("Valeur lue: %x\n", var);
#else
    int fd= open("plop.bin", O_WRONLY | O_CREAT, 0644);
    assert(fd >= 0);
    assert(write(fd, &var, sizeof(var)) >= 0);
#endif /* READER */
    close(fd);
    return 0;
}
```


Endianness

- Sous Mac OS X (Intel Core Duo)

```
mortimer bqu$ gcc -Wall -Werror -o plop plop.c
mortimer bqu$ gcc -Wall -Werror -DREADER -o plop_read plop.c
mortimer bqu$ ./plop
mortimer bqu$ ls -l
-rw-r--r--  1 bqu      bqu      4 Feb 16 11:43 plop.bin
mortimer bqu$ hexdump -C plop.bin
00000000  21 87 ba fe                |!...|
00000004
mortimer bqu$ ./plop_read
Valeur lue: feba8721
mortimer bqu$
```

Endianness

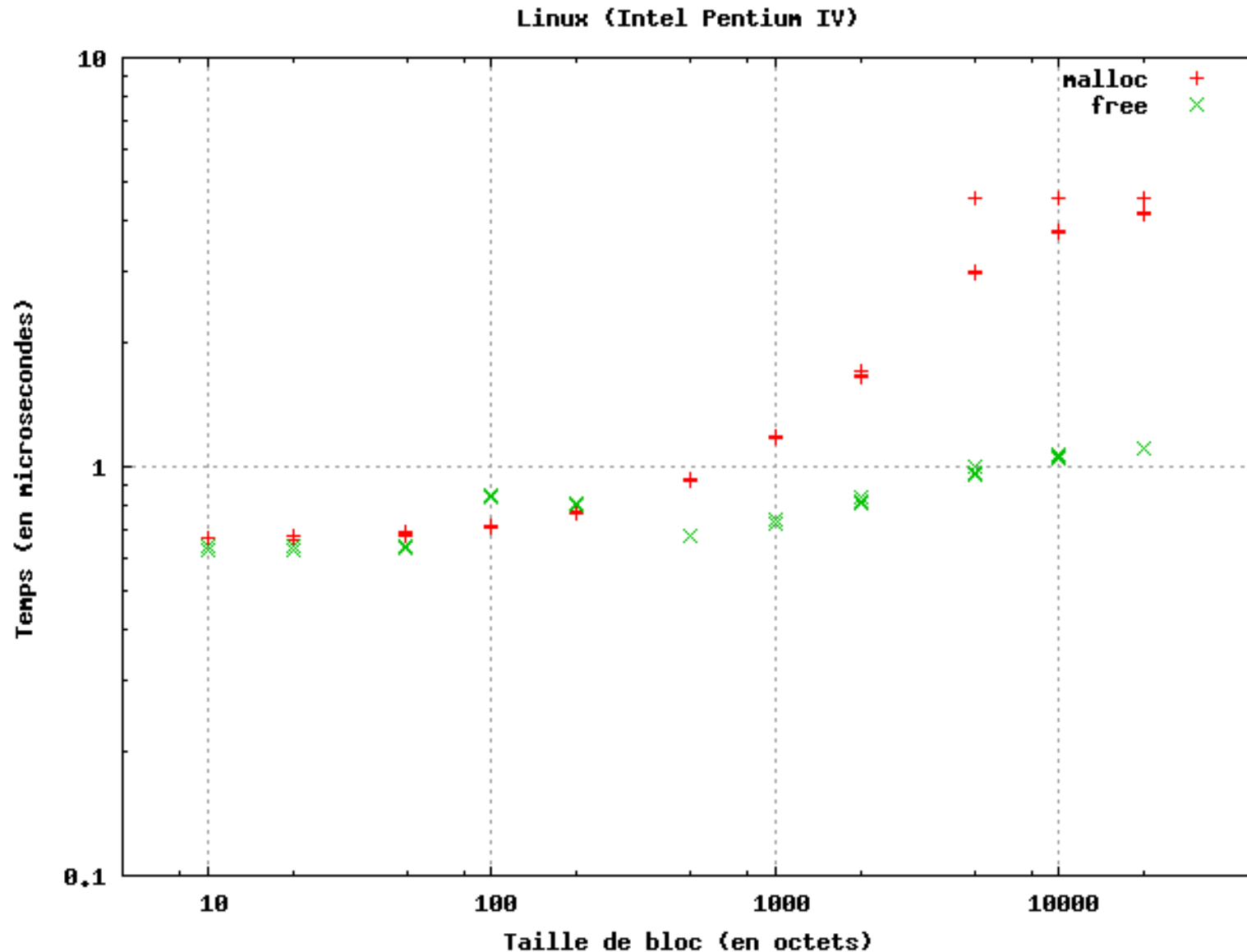
- Copie sur Sirius (UltraSPARC-III)

```
mortimer bqu$ scp plop.c bqu@sirius.info.ucl.ac.be:/tmp
mortimer bqu$ scp plop.bin bqu@sirius.info.ucl.ac.be:/tmp
mortimer bqu$
```

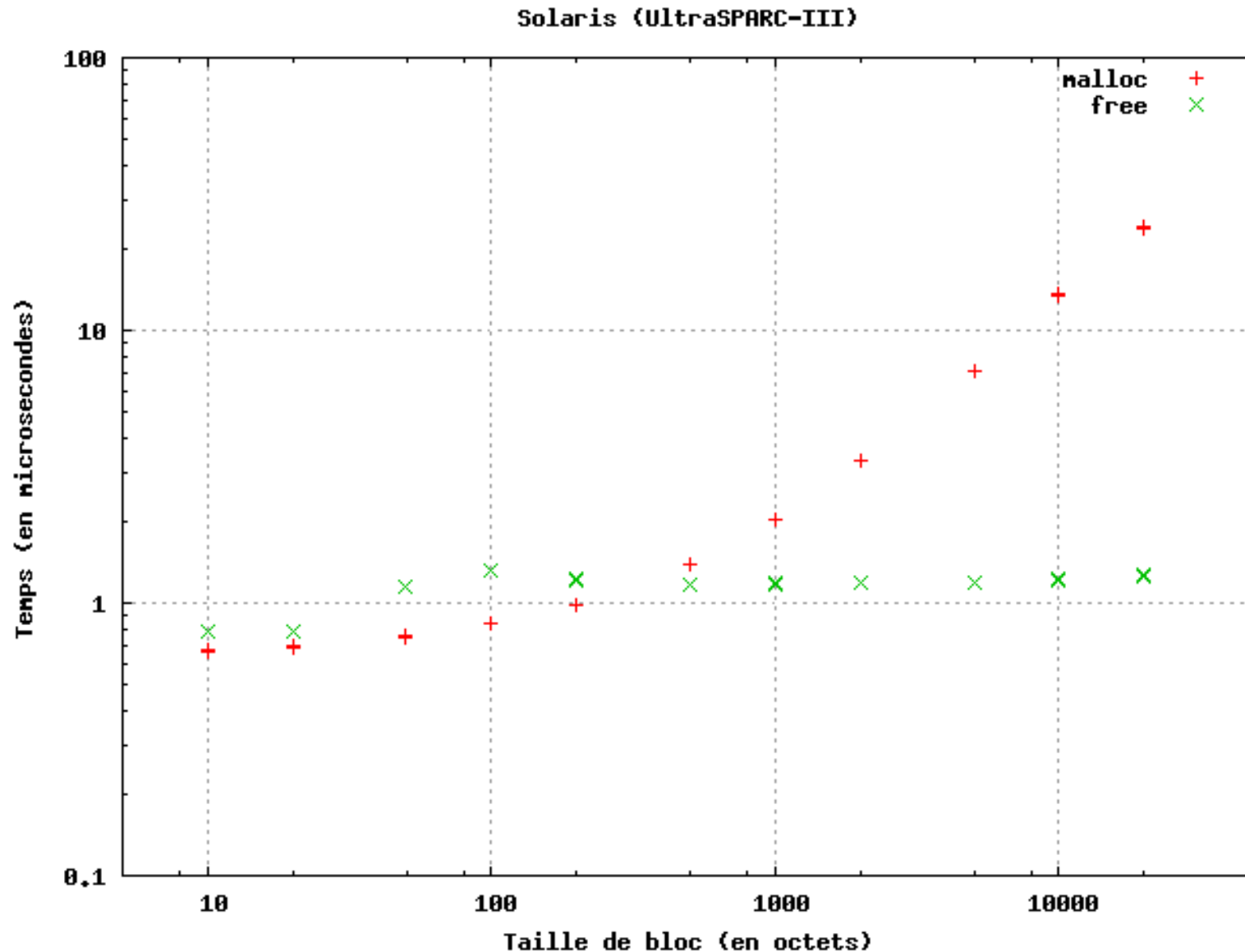
```
-bash-3.00$ gcc -Wall -Werror -DREADER -o plop_read plop.c
-bash-3.00$ ./plop
-bash-3.00$ ls -l
-rw-r--r--  1 bqu      stafinfo      4 Feb 16 11:50 plop.bin
-bash-3.00$ /opt/csw/bin/hexdump -C plop.bin
0000  21 87 ba fe                                     !...
-bash-3.00$ ./plop_read
Valeur lue: 2187bafe
-bash-3.00$
```

Malloc() is not free

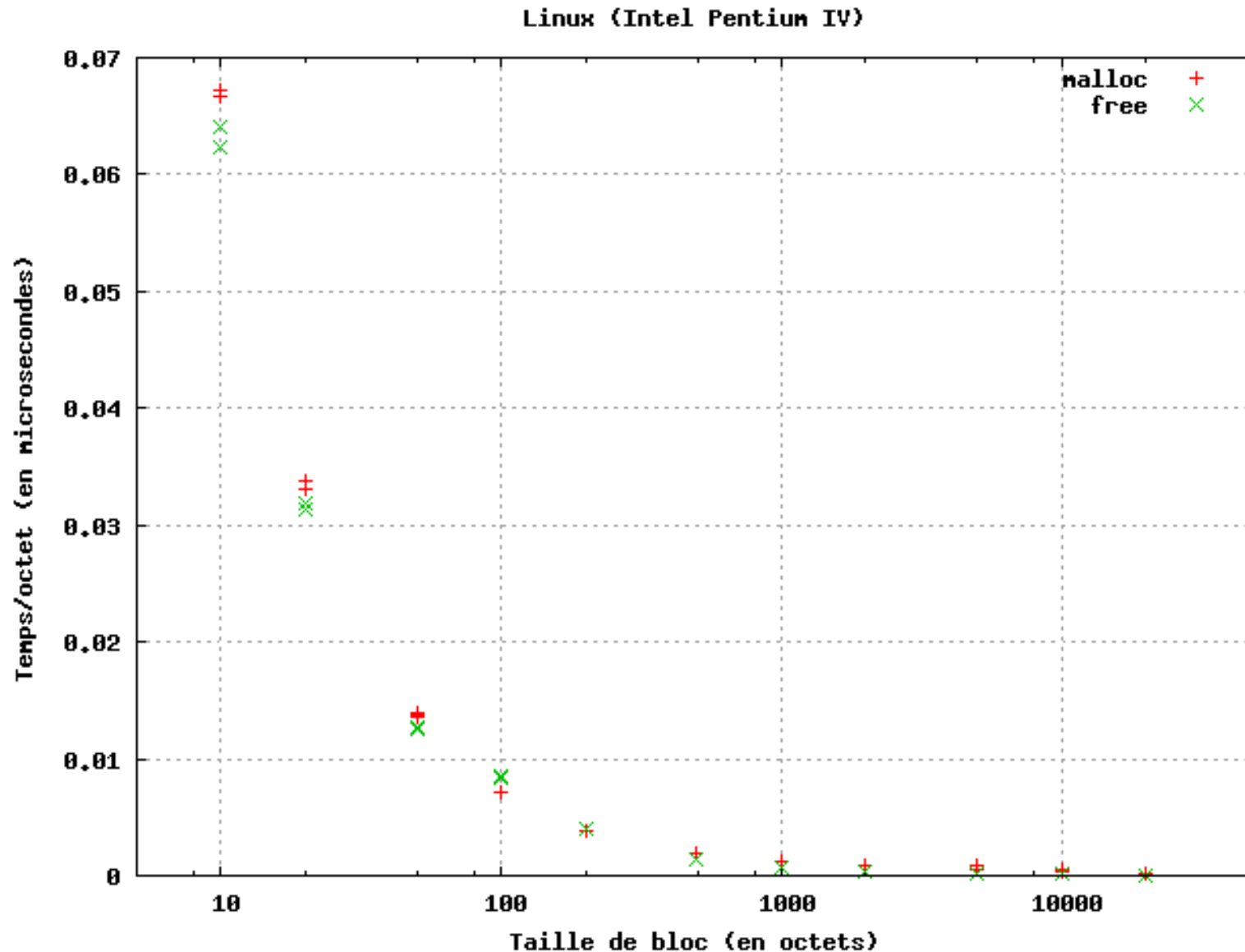
Malloc() is not free



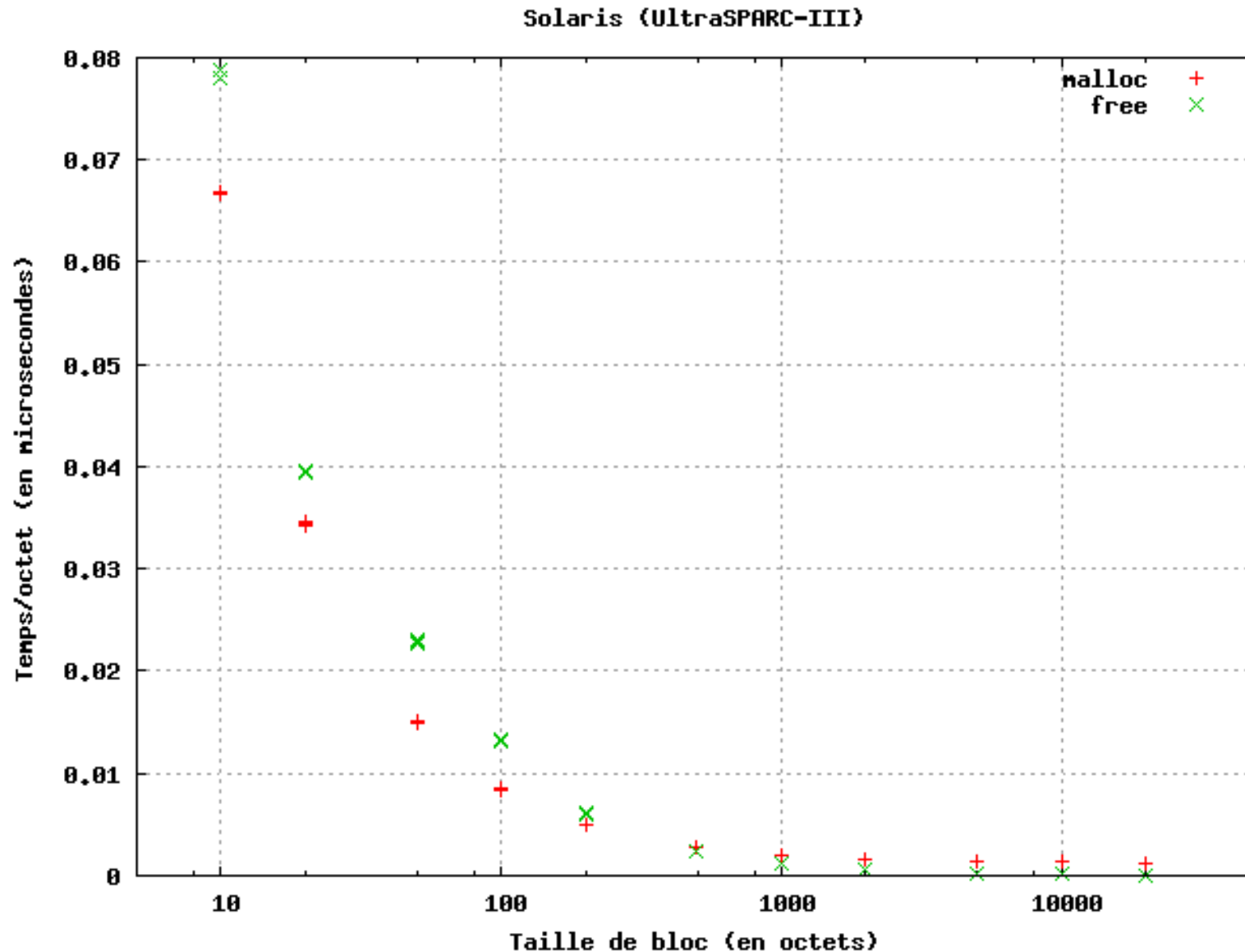
Malloc() is not free



Malloc() is not free



Malloc() is not free



Suggestion d'exercices

Exercice 1

- Impression d'un nombre

```
#ifndef __PRINT_NUM_H__
#define __PRINT_NUM_H__

typedef enum {
    BASE_BIN, BASE_OCT, BASE_DEC, BASE_HEX
} base_t;

/**
 * Affiche un nombre dans la base donnée.
 * \param num      est le nombre à afficher
 * \param output_base est la base utilisée pour l'affichage
 */
int print_num(unsigned int num, base_t output_base);

#endif /* __PRINT_NUM_H__ */
```

Exercice 2

- Bubble Sort

```
#ifndef __BUBBLE_SORT_H__
#define __BUBBLE_SORT_H__

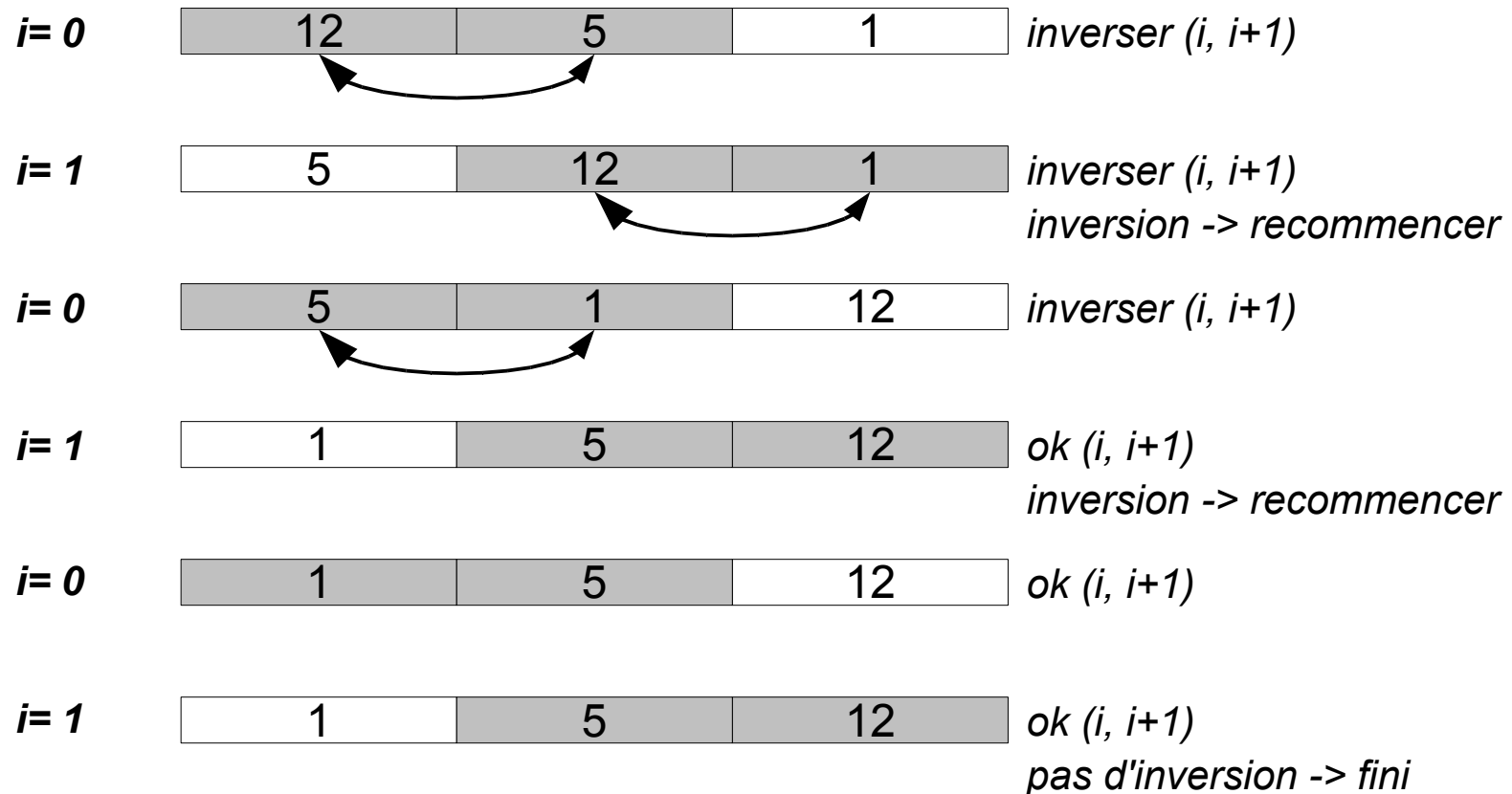
/** Type fonction de comparaison.
 * \param e1 entier à comparer
 * \param e2 entier à comparer
 * \return -1 si (e1<e2), 1 si (e1>e2) et 0 si (e1=e2) */
typedef int (*cmp_t)(int e1, int e2);

/** Trie un tableau avec l'algorithme Bubble Sort
 * \param tab est le tableau à trier
 * \param size est la taille du tableau
 * \param cmp_fct est la fonction de comparaison à utiliser
 */
int bubble_sort(int tab[], unsigned int size, cmp_t cmp_fct);

#endif /* __PRINT_NUM_H__ */
```

Exercice 2

- Bubble Sort



Note: D'un point de vue algorithmique, Bubble Sort est un mauvais exemple !!! mais il est facile à implémenter.

Exercice 3

- Mesure de performance lecture fichier

```
bqu@mortimer$ dd bs=1k count=10000 if=/dev/urandom  
of=random_medium.bin  
bqu@mortimer$
```

```
unsigned char buffer[  ];  
int num_read;  
int fd= open("random_medium.bin", O_RDONLY, 0);  
assert(fd >= 0);  
do {  
    /* garder temps début */  
    num_read= read(fd, buffer, block_size);  
    /* prendre temps fin, calculer différence */  
} while (num_read > 0);  
/* afficher temps total */  
close(fd);
```

Exercice 3

- Mesure de performance lecture fichier

```
gettimeofday()  
  
timerclear()  
timeradd()  
timersub()  
  
getrusage()  
  
struct timeval {  
    time_t      tv_sec;  
    suseconds_t tv_usec;  
};  
struct timeval tv_diff= ...;  
printf("Différence de temps: %ld.%.6ld\n",  
       tv_diff.tv_sec, tv_diff.tv_usec);
```

Exercice 3

- Mesure de performance lecture fichier

```
1      TEMPS (1)
2      TEMPS (2)
5      TEMPS (5)
10     TEMPS (10)
20     TEMPS (20)
50     TEMPS (50)
100    TEMPS (100)
200    TEMPS (200)
500    TEMPS (500)
1000   TEMPS (1000)
2000   TEMPS (2000)
5000   TEMPS (5000)
10000  TEMPS (10000)
20000  TEMPS (20000)
```

Exercice 3

- Afficher graphique avec gnuplot

```
set logscale x
set grid
set xlabel "Taille de bloc (octets)"
set ylabel "Temps"
plot "perf_file.result" u 1:2 w l t "gettimeofday()", \
    "" u 1:3 w l t "user time", \
    "" u 1:4 w l t "system time"
pause -1
```

```
bqu@mortimer$ gnuplot perf_file.gnuplot
bqu@mortimer$
```

FIN

Questions ?

Remerciements: Merci à Sébastien Barré pour ses commentaires sur cette partie du cours. Merci à S. Marquet pour ses questions éclairantes/inspirantes. Merci à Y. Gillet pour ses commentaires.