Pratique du C Classes d'allocation, paramètres par la

V-2 (09-03-2012)

Pratique du C d'allocation. contexte et passage de paramètres par la pile

Les classes d'allocation des

Passage de

V74 (09-03-2012)

Pratique du C Classes d'allocation, contexte et passage de paramètres par la pile

Les classes d'allocation des

Pratique du C Classes d'allocation, contexte et passage de paramètres par la pile

Licence Informatique — Université Lille 1 Pour toutes remarques : Alexandre.Sedoglavic@univ-lille1.fr

Semestre 5 — 2012-2013

www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf V74 (09-03-2012)

Les variables externes (extern) sont :

- ▶ allouées en zone statique de données (dans un segment de données):
- allouées à la compilation (valeur par défaut 0);
- durée de vie du programme;
- visibles depuis toutes les fonctions.

Les variables statiques (static) sont :

- allouées comme les variables externes;
- ▶ et si elles sont définies :
 - ▶ à l'extérieur de toute fonction, elles sont visibles depuis les fonctions déclarées dans le fichier source les
 - ▶ à l'intérieur d'une fonction, elles sont visibles depuis la fonction seulement, mais reste allouées en dehors de l'exécution de la fonction (valeur conservée entre les différents appels).

4 D F 4 B F 4 E F 4 E F 90 P www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf V74 (09-03-2012)

Par exemple, on peut avoir le code suivant :

```
int global = 1; /* d\'efinition d'une variable
                  externe (globale)
extern int extern_global ; /* d\'eclaration d'une variable
                 globale d'un autre fichier (externe) */
```

static int global_privee = 2 ; /* globale au fichier, invisible depuis d'autres fichiers (statique) */

fonction(int param) {/* param\'etre (automatique) */

auto int local = 3; /* variable automatique (locale) */ /* le mot clef auto ne sert \'a rien */ static int local_static = 4 ; /* variable statique (locale) valeur inchang\'ee entre 2 appels */

register int i = 5; /* variable register (locale) */

return i++ ;

Pratique du C Classes d'allocation, contexte et passage de paramètres par la pile

Les classes d'allocation des

Pratique du C Classes d'allocation contexte et passage de paramètres par la

pile

Les classes d'allocation des

Passage de

Pratique du C Classes d'allocation, contexte et paramètres par la pile

variables

Les classes d'allocation des variables

En C, les variables ont pour attribut :

- leur nom : un identificateur;
- leur type : type de base ou défini par l'utilisateur;
- une classe d'allocation indiquant :
 - le type de l'emplacement mémoire où est allouée la variable;
 - sa durée de vie :
 - > sa visibilité par les différentes fonctions.

Il y a 5 classes d'allocation :

```
externe automatique statique "register" volatile
```

```
www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf
```

Les variables automatiques (auto) sont :

- ▶ allouées dynamiquement sur la pile (valeur initiale indéterminée);
- allouées à chaque entrée dans la fonction ou le bloc où la variable est définie (paramètres, variables locales);
- durée de vie de la fonction ou du bloc;
- visibles uniquement depuis la fonction ou le bloc.

Les variables de registres (register) sont :

- allouées si possible dans un registre du processeur;
- des variables de type simple uniquement;
- des variables de classe automatique uniquement;
- et ne possèdent pas d'adresse.

4 D F 4 D F 4 E F 4 E F 9 9 9 9 www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf

Le code assembleur correspondant est :

```
int global = 1;
                                       .data
                                .globl global
                               global:
extern int extern_global ;
                                       .long
static int global_privee = 2 ;
                               global_privee:
                                       .long
int fonction(int param) {
                               local static.0:
                                       .long 4
  int local = 3;
  static int local_static = 4 ; .globl fonction
                               fonction:
  register int i = 5;
                                     pushl %ebp
                                     movl
                                            %esp, %ebp
  return i++ ;
                                     subl
                                            $4, %esp
                                            $3, -4(%ebp)
                                     movl
                                     movl
                                            $5, %eax
                                     leave
```

```
Pratique du C
Classes
d'allocation,
                                                                                       Pratique du C
Classes
d'allocation,
                                                                                                      Les variables volatiles sont susceptibles d'être modifiées
                                                                                                      indépendamment du code les déclarant. Considérons
                extern int ailleurs(int) ;
                                                             .text
                                                                                        contexte et
                                                             .type
                                                                     foo,@function
                                                                                                      l'exemple suivant (seul dans un fichier sources) :
 paramètres par l
                static int foo(void){
                                                     foo:
                                                                                       paramètres par la
                                                                                           pile
                         return 1 ;
                                                             pushl
                                                                      %ebp
                                                                                                      static int foo; /* foo pourrait \^etre un pointeur sur
                                                                      %esp, %ebp
                                                                                                                           un segment de m\'emoire partag\'ee */
                                                            movl
                                                            movl
                                                                     $1, %eax
                int bar(void){
                                                            popl
                                                                     %ebp
                                                                                                      bar(void){
                         return 1 ;
                                                             ret
                                                                                                         foo=0;
Effet des mots clef
                                                                                      Effet des mots clei
                                                                                                         while(foo!=1)
static et extern
sur les fonctions
                                                    .Lfe1:
                                                                                      static et extern
                                                                                      sur les fonctions
                                                             .size
                                                                     foo..Lfe1-foo
                                                                                                           continue :
                                                    .globl bar
                /* la fonction ailleurs
                                                                     bar,@function
                                                             .type
                                                                                                      Un compilateur en optimisant ce code remplacera la boucle
                   est d\'eclar\'ee mais
                                                    har:
                   d\'efinie dans un autre
                                                            pushl
                                                                     %ebp
                                                                                                      par while(1) car la classe d'allocation static lui assure que
                                                            movl
                   fichier source (i.e. object).
                                                                     %esp, %ebp
                                                                                                      seule la fonction bar peut modifier foo. Cependant,
                                                            movl
                                                                     $1. %eax
                                                                                                      l'emplacement mémoire associé à la variable pourrait être
                   la fonction foo n'est pas
                                                            popl
                                                                     %ebp
                                                                                                      partagé — et donc modifiable — par un autre processus
                   accessible depuis un autre
                                                             ret
                                                             .Lfe2:
                   fichier alors que bar l'est.
                                                                                                      (voir l'unité d'enseignement Pratique des systèmes) et
                                                             .size bar, .Lfe2-bar
                                                                                                      l'optimisation du compilateur ne pas correspondre à la
                                                                                                      volonté du programmeur.
V74 (09-03-2012)
                                                      www.fil.univ-lille1.fr/~sedoglay/C/Cours09.pdf V74 (09-03-2012)
                                                                                                                                            www.fil.univ-lille1.fr/~sedoglay/C/Cours09.pdf
                L'usage de la classe d'allocation volatile supprime
                                                                                                      Une variable dont le type est qualifié par const ne peut pas
  Pratique du C
                                                                                       Pratique du C
                l'optimisation concernant la variable ainsi qualifiée. Le code :
                                                                                                      être modifiée.
  d'allocation
                                                                                        d'allocation.
  contexte et
                                                                                        contexte et
                                                                                                      Ce qualificatif permet au programmeur de s'assurer de ne
passage de
paramètres par la
                                                                                       passage de
paramètres par la
                static volatile int foo;
                                                                                                      pas modifier des variables passées par référence comme par
     pile
                                                                                           pile
                void
                                                                                                      dans le cas suivant :
                bar
                (void)
                                                                                                      #include <string.h>
                                                                                                      int strcmp(const char *s1, const char *s2);
                   foo=0 :
Effet des mots clef
                                                                                      Effet des mots clef
                                                                                                      Ainsi, on est sur que cette fonction ne va pas modifier les
                   while(foo!=1)
sur les fonctions
                                                                                      sur les fonctions
                     continue :
                                                                                                      chaînes de caractères passées en arguments.
                   return ;
                                                                                                      Ce modificateur de type impose la définition au moment de
                }
                                                                                                      la déclaration (const int a=0; et pas const int a; a=0;).
                correspond donc au cas de figure au la variable foo est
                                                                                                      Remarquons que les qualificateurs de types peuvent être
                partagée (par exemple entre deux processus légers ou dans
                                                                                                      utilisés finement :
                un segment de mémoire partagé — cf. cours Pratique des
                                                                                                                                  /* caract\'ere constant */
                                                                                                         const char c ;
                Systèmes).
                                                                                                         const char *s ; /* pointeur vers caract\'eres constants */
Passage de
                Sans volatile, le compilateur simplifie ce code en boucle
                                                                                                         char const *s ; /* pointeur constant vers caract\'eres */
                                                                                                         const char * const s ; /* pointeur constant
                infinie (comme foo n'est pas modifié, cette variable est
                                                                                                                                      vers caract\'eres constants */
                supprimée).
V74 (09-03-2012)
                                                      www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf V74 (09-03-2012)
                                                                                                                                            www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf
                                                                                                      Il est possible de déclarer une fonction comme ayant un
                                                                                       Pratique du C
  Pratique du C
                Attention aux surprises lors de l'évaluation des paramètres à
  Classes
d'allocation,
                                                                                        Classes
d'allocation,
                                                                                                      nombre variable de paramètres en « déclarant » les
                transmettre:
  contexte et
                                                                                        contexte et
                                                                                                      paramètres optionnels par l'unité lexicale ... (3 points à la
                                                                                        passage de
 paramètres par la
                                                                                       paramètres par la
                                                                                                      suite) :
                #include <stdio.h>
                                             rodata
     pile
                                                                                          pile
                                      .LCO: .string "le premier argument %c..."
                                                                                                      int
                int main(void){
                                            .text .globl main
                                                 main: pushl
                                                                   %ebp
                                                                                                      (char *par_obl, ...)
                   int foo = 'a';
                                                          movl
                                                                   %esp, %ebp
                                                          subl
                                                                   $8, %esp
                                                                                                         return 0 ;
                   printf("le premier argument %c
                                                                   $97, -4(%ebp)
                                                          movl
                      et le second %c\n",foo,foo++); subl
                                                                   $4, %esp
Ordre d'évaluation
                   return 0 :
                                                                   -4(%ebp), %eax
                                                          movl
                                                                                                      Une fonction peut avoir à la fois des paramètres obligatoires
des paramètres
                }
                                                                   %eax
                                                          pushl
                                                                                                      et des paramètres optionnels, les paramètres obligatoires
                                                                                      Fonction à
                                                                   -4(%ebp), %eax
                                                                                      nombre variable
                                                          leal
                                                                                      de paramètres
                                                                                                      apparaissant en premier et l'unité lexicale ... apparaissant en
                                                          incl
                                                                   (%eax)
                                                                                                      dernière position dans la liste de déclaration des paramètres
                                                                   -4(%ebp)
                le premier argument b et le second a pushl
                                                                                                      formels.
                                                          pushl
                                                                   $.LCO
                                                          call
                                                                   printf
                                                                                                      Généralement, un paramètre obligatoire indique le nombre et
                                                                   $16, %esp
                                                          addl
                                                                                                      le type des paramètres optionnels comme dans le cas de
                                                          movl
                                                                  $0, %eax
                                                                                                      printf :
                                                          leave ret
                                                                                                      printf("premier argument %c et second %c\n",foo,foo++) ;
                                                       10 1 10 12 12 12 1 2 99 C
```

www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf V74 (09-03-2012)

www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf

V74 (09-03-2012)

```
On appelle contexte d'un appel de fonction dans la pile
 Pratique du C
Classes
d'allocation,
                                                                                      Pratique du C
Classes
d'allocation,
               Un exemple d'utilisation (paramètres
                                                                                                     d'exécution la partie de la pile associée :
              optionnels de même type)
  contexte et
                                                                                       contexte et
passage de
paramètres par la
pile
                                                                                                        paramètres d'appels;
                                                                                      paramètres par la
                                                                                          pile
                int somme(int nbpar, ...){

    variables automatiques de la fonction.

                  int *pt = &nbpar ; /* on fait pointer pt
                                          sur le premier param\'etre */
                  int res = 0:
                                                                                                        backtrace : les contextes disponibles
                  for(;nbpar>0;nbpar--){
                                 ; /* on passe au param\'etre suivant */
                                                                                                          (gdb) backtrace
                      pt++
                      res += *pt ;
Fonction à
                    }
                                                                                                          #1 0x08048532 in main () at SMS.c:45
 ombre variable
                                                                                                          #2 0x400327f7 in __libc_start_main
de paramètres
                  return res ;
               }
                                                                                     Définition : un
                                                                                                        ▶ info frame nb : qui affiche un contexte
                                                                                      contexte dans la
                                                                                     pile d'exécution
                                                                                                          (gdb) info frame 2
                                                                                      (stack frame)
                main(void){
                  return somme(3,1,2,3)+somme(4,4,5,3,1);
                                                                                                          Arglist at 0xbffff6f8, args:
                                                                                                          Locals at Oxbfffff6f8, Previous frame's sp in esp
                                                                                                           Saved registers: ebp at 0xbffff6f8, eip at 0xbffff6fc
                                                      ARKARIATE TOOR
V74 (09-03-2012)
                                                      www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf V74 (09-03-2012)
                On peut étendre la notion de contexte en y associant en plus
 Pratique du C
                                                                                      Pratique du C
                des informations concernant la pile d'exécution, l'état des
  d'allocation
                                                                                       d'allocation.
  contexte et
                                                                                       contexte et
                registres du processeur (%EAX, %EIP, %CS, etc).
                                                                                      passage de
paramètres par la
passage de
paramètres par la
                                                                                                     des systèmes (signal).
                Ce faisant, on peut faire des branchements non-locaux (i.e.
    pile
                des branchements à un endroit presque arbitraire du code)
                sans utiliser goto. Deux fonctions de la librairie standard sont
                dédiées à cet effet. Schématiquement,
                  setjmp mémorise son contexte juste avant son RET;
                                                                                                     registres le sont). Sur une architecture de type intel, on a :
                  ▶ longjmp permet de rétablir le contexte mémorisé en
                                                                                                     # if __WORDSIZE == 64
                     plaçant son second argument dans %EAX.
                                                                                                     typedef long int __jmp_buf[8];
                #include <setjmp.h>
                #include <stdio.h>
                                                                                                     typedef int __jmp_buf[6];
                int main (void){
                                                                                                     # endif
                                                                                                     struct __jmp_buf_tag{
                int i = setjmp(env); /* au premier appel setjmp retourne 0 */
                                                                                                      __jmp_buf __jmpbuf;
                printf("i = %d\n",i);
setjmp/longjmp
                                                                                      setimp/longimp
                if(i == 2) return 0 ;
                longjmp(env,2); /* on branche sur setjmp qui retourne 2 */
                                                                                                     };
                return 1 ; /* cette instruction n'est jamais ex\'ecut\'ee */
                                                                                                     typedef struct __jmp_buf_tag jmp_buf[1];
V74 (09-03-2012)
                                                      www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf V74 (09-03-2012)
                                                                                      Pratique du C
  Pratique du C
               Passage de paramètres par copie : une
 Classes
d'allocation,
                                                                                       d'allocation,
               copie est faite sur la pile
  contexte et
                                                                                       contexte et
                                                                                                                                        .text
 paramètres par la
                                                                                      paramètres par la
                                                                                                                                        .globl main
    pile
                                                                                         pile
                                                                                                                                        main:
                                               .globl main
                                                main:
                                                   pushl %ebp
                                                   movl %esp, %ebp
                                                   subl $8, %esp
                                                   andl $-16, %esp
                                                   movl $1, -4(%ebp)
                                                                                                     int main(void){
                                                   movl $1, -8(%ebp)
                                                   subl $8, %esp
                int main(void){
                                                                                                             int a = 1;
                                                                                                             int b = 1:
                        int a = 1;
                                                                                                             PER(a,b);
                        int b = 1;
                                                                                                              return 0 :
                                                                                                                                           leave
                         return 0 ;
                                                   movl $0, %eax
               }
                                                   leave
```

Passage de

Passage de

V74 (09-03-2012)

▶ adresse de retour et ancien pointeur de contexte %EBP;

Les différentes portions de pile correspondant aux différents contextes d'exécution peuvent être obtenues dans gdb par :

```
#0 traduction (code=0xbffff660 "oeu") at SMS.c:12
Stack frame at 0xbffff6f8: eip = 0x400327f7 in _main;
saved eip 0x8048301 caller of frame at 0xbffff6d8
```

www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf

Dans l'exemple précédent, set jmp et long jmp sont utilisées dans la même fonction mais généralement, ces fonctions sont utilisées pour la gestion d'erreurs et la programmation

Limitation: longjmp ne permet pas de revenir à n'importe quel point mémorisé par l'appel setjmp; ce n'est possible que si la fonction qui a exécutée le setjmp(env) n'est pas terminée car l'état de la pile n'est pas mémorisé (seuls les

```
/* pour stocker les registres */
  int __mask_was_saved; /* pour d'autres usages syst\'eme */
 __sigset_t __saved_mask;/* li\'es aux signaux (cf. PDS) */
extern int setjmp (jmp_buf __env) ;
                                 www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf
```

pushl %ebp movl %esp, %ebp subl \$8, %esp andl \$-16, %esp movl \$1, -4(%ebp) movl \$1, -8(%ebp) subl \$8, %esp pushl -8(%ebp) pushl -4(%ebp) call PER addl \$16, %esp movl \$0, %eax ret

```
Pratique du C
Classes
d'allocation,
    Pratique du C
Classes
d'allocation,
                                                                                                                                                                                                                                                 contexte et
  passage de
paramètres par la
pile
                                                                                                                                                                                                                                             paramètres par la
pile
                                                                                                                                                         .text
                                                                                                                                                                                                                                                                                                                                                                                                   .text
                                                                                                                                                   .globl PER
                                                                                                                                                                                                                                                                                                                                                                                             .globl PER
                                                                                                                                               PER:
                                             void PER(int alpha, int beta){    pushl %ebp
                                                                                                                                                                                                                                                                                       void PER(int alpha, int beta){
                                                                                                                                                                                                                                                                                                                                                                                             pushl %ebp
                                                                 int tmp = alpha ;
                                                                                                                                                   movl %esp, %ebp
                                                                                                                                                                                                                                                                                                             int tmp = alpha ;
                                                                                                                                                                                                                                                                                                                                                                                               movl %esp, %ebp
                                                                                                                                                   subl $4, %esp
                                                                                                                                                                                                                                                                                                             alpha = beta ;
                                                                                                                                                                                                                                                                                                                                                                                               subl $4, %esp
                                                                                                                                                   movl 8(%ebp), %eax
                                                                                                                                                                                                                                                                                                                                                                                               movl 8(%ebp), %eax
                                          }
                                                                                                                                                   movl %eax, -4(%ebp)
                                                                                                                                                                                                                                                                                                                                                                                               movl %eax, -4(%ebp)
                                                                                                                                                                                                                                                                                                                                                                                               movl 12(%ebp), %eax
                                                                                                                                                                                                                                                                                                                                                                                               movl %eax, 8(%ebp)
                                                                                                                                                 leave
                                                                                                                                                                                                                                                                                                                                                                                               leave
                                                                                                                                                 ret
                                                                                                                                                                                                                                                                                                                                                                                               ret
Passage de
                                                                                                                                                                                                                                            Passage de
paramètres
                                                                                                                                                       4 D > 4 D > 4 E > 4 E > E 994 P
                                                                                                                                                                                                                                                                                                                                                                                                 40 × 40 × 42 × 42 × 2 900
V74 (09-03-2012)
                                                                                                                                                                                                                                                                                                                                                                                                www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf
                                                                                                                                                      www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf V74 (09-03-2012)
     Pratique du C
                                                                                                                                                                                                                                               Pratique du C
Classes
                                                                                                                                                                                                                                                                                    Passage de paramètre par adresse : les
     d'allocation
                                                                                                                                                                                                                                                d'allocation
                                                                                                                                                                                                                                                                                    adresses sont copiées sur la pile
      contexte et
                                                                                                                                                                                                                                                 contexte et
 passage de
paramètres par la
                                                                                                                                                                                                                                            passage de
paramètres par la
            pile
                                                                                                                                                         .text
                                                                                                                                                                                                                                                       pile
                                                                                                                                                                                                                                                                                                                                                                                                  text
                                                                                                                                                   .globl PER
                                                                                                                                                                                                                                                                                                                                                                                                   .globl main
                                                                                                                                               PER:
                                                                                                                                                                                                                                                                                                                                                                                        main:
                                             void PER(int alpha, int beta){
                                                                                                                                                   pushl %ebp
                                                                                                                                                                                                                                                                                                                                                                                                 pushl %ebp
                                                                 int tmp = alpha ;
                                                                                                                                                   movl %esp, %ebp
                                                                                                                                                                                                                                                                                                                                                                                                 movl %esp, %ebp
                                                                 alpha = beta ;
                                                                                                                                                    subl $4, %esp
                                                                                                                                                                                                                                                                                                                                                                                                 subl $8, %esp
                                                                 beta = tmp ;
                                                                                                                                                   movl 8(%ebp), %eax
                                                                                                                                                                                                                                                                                                                                                                                                 andl $-16, %esp
                                                                                                                                                   movl %eax, -4(%ebp)
                                                                                                                                                                                                                                                                                                                                                                                                 movl $1, -4(%ebp)
                                                                                                                                                   movl 12(%ebp), %eax
                                                                                                                                                                                                                                                                                                                                                                                                 movl $1, -8(%ebp)
                                                                                                                                                   movl %eax. 8(%ebp)
                                                                                                                                                                                                                                                                                                                                                                                                 subl $8, %esp
                                                                                                                                                                                                                                                                                       int main(void){
                                                                                                                                                   movl -4(%ebp), %eax
                                                                                                                                                   movl %eax, 12(%ebp)
                                                                                                                                                                                                                                                                                                               int a = 1;
                                                                                                                                                   leave
                                                                                                                                                                                                                                                                                                              int b = 1;
                                                                                                                                                   ret
                                                                                                                                                                                                                                                                                                                return 0;
                                                                                                                                                                                                                                                                                       }
Passage de
                                                                                                                                                                                                                                            Passage de
                                                                                                                                                                                                                                                                                                                                                                                                  movl $0, %eax
                                                                                                                                                                                                                                                                                                                                                                                                 leave
                                                                                                                                                       4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 4 m > 
                                                                                                                                                                                                                                                                                                                                                                                                        1) (A) (A) (B) (B) (A) (A)
V74 (09-03-2012)
                                                                                                                                                      www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf V74 (09-03-2012)
                                                                                                                                                                                                                                                                                                                                                                                                 www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf
     Pratique du C
                                                                                                                                                                                                                                               Pratique du C
     Classes
d'allocation,
                                                                                                                                                                                                                                                Classes
d'allocation,
      contexte et
                                                                                                                                                              .text
                                                                                                                                                                                                                                                 contexte et
                                                                                                                                                                                                                                                                                                                                                                                                                     .text
 passage de
paramètres par la
                                                                                                                                                                                                                                            passage de
paramètres par la
                                                                                                                                                               .globl main
                                                                                                                                                                                                                                                                                                                                                                                                 .globl PER
            pile
                                                                                                                                                 main:
                                                                                                                                                                                                                                                        pile
                                                                                                                                                                                                                                                                                                                                                                                               PER:
                                                                                                                                                          pushl %ebp
                                                                                                                                                                                                                                                                                       void PER(int *alpha, int *beta){     push1 %ebp
                                                                                                                                                          movl %esp, %ebp
                                                                                                                                                                                                                                                                                                             int tmp = *alpha ;
                                                                                                                                                                                                                                                                                                                                                                                                    movl %esp, %ebp
                                                                                                                                                          subl $8, %esp
                                                                                                                                                                                                                                                                                                                                                                                                     subl $4, %esp
                                                                                                                                                          andl $-16, %esp
                                                                                                                                                                                                                                                                                                                                                                                                     movl 8(%ebp), %eax
                                                                                                                                                         movl $1, -4(%ebp)
                                                                                                                                                                                                                                                                                                                                                                                                    movl (%eax), %eax
                                                                                                                                                         movl $1, -8(%ebp)
                                                                                                                                                                                                                                                                                                                                                                                                     movl %eax, -4(%ebp)
                                                                                                                                                          subl $8, %esp
                                            int main(void){
                                                                                                                                                         leal -8(%ebp), %eax
                                                                     int a = 1;
                                                                                                                                                          pushl %eax
                                                                     int b = 1;
                                                                                                                                                         leal -4(%ebp), %eax
                                                                                                                                                          pushl %eax
                                                                    PER(&a,&b) ;
                                                                    return 0;
                                                                                                                                                          call PER
                                           }
                                                                                                                                                          addl $16, %esp
                                                                                                                                                         movl $0, %eax
                                                                                                                                                         leave
                                                                                                                                                                                                                                                                                                                                                                                                     leave
Passage de
                                                                                                                                                                                                                                            Passage de
                                                                                                                                                          ret
                                                                                                                                                                                                                                                                                                                                                                                                     ret
                                                                                                                                                                                                                                                                                                                                                                                                 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 4 m x 
                                                                                                                                                       40 × 40 × 42 × 42 × 2 990
V74 (09-03-2012)
                                                                                                                                                      www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf V74 (09-03-2012)
                                                                                                                                                                                                                                                                                                                                                                                                 www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf
```

```
Pratique du C
Classes
d'allocation,
                                                                                         Pratique du C
Classes
d'allocation,
                                                               .text
                                                                                                                                                      .text
  contexte et
                                                                                         contexte et
passage de
paramètres par la
pile
                                                        .globl PER
                                                                                                                                               .globl PER
                                                                                        paramètres par la
                                                       PER:
                                                                                            pile
                                                                                                                                              PER:
                void PER(int *alpha, int *beta){
                                                        pushl %ebp
                                                                                                       void PER(int *alpha, int *beta){
                                                                                                                                               pushl %ebp
                        int tmp = *alpha ;
                                                         movl %esp, %ebp
                                                                                                                int tmp = *alpha ;
                                                                                                                                                movl %esp, %ebp
                        *alpha = *beta ;
                                                         subl $4, %esp
                                                                                                                *alpha = *beta;
                                                                                                                                                subl $4, %esp
                                                         movl 8(%ebp), %eax
                                                                                                                *beta = tmp ;
                                                                                                                                                movl 8(%ebp), %eax
                }
                                                         movl (%eax), %eax
                                                                                                                                                movl (%eax), %eax
                                                         movl %eax, -4(%ebp)
                                                                                                                                                      %eax, -4(%ebp)
                                                         movl 8(%ebp), %edx
                                                                                                                                                movl 8(%ebp), %edx
                                                         movl 12(%ebp), %eax
                                                                                                                                                movl
                                                                                                                                                      12(%ebp), %eax
                                                         movl (%eax), %eax
                                                                                                                                                movl
                                                                                                                                                       (%eax), %eax
                                                         movl %eax, (%edx)
                                                                                                                                                      %eax, (%edx)
                                                                                                                                                movl
                                                                                                                                                movl 12(%ebp), %edx
                                                                                                                                                movl -4(%ebp), %eax
                                                                                                                                                movl (%eax), %eax
                                                                                                                                                movl %eax, (%edx)
                                                         leave
                                                                                                                                                leave
Passage de
                                                                                        Passage de
                                                         ret
                                                                                                                                                ret
paramètres
                                                        40 × 40 × 43 × 43 × 9 × 90 0
                                                                                                                                               4 D > 4 B > 4 B > 4 B > 9 4 P 9
V74 (09-03-2012)
                                                                                                                                               www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf V74 (09-03-2012)
 Pratique du C
                                                                                         Pratique du C
               Passage de paramètre de type structure
  d'allocation
                                                                                         d'allocation
  contexte et
                                                                                         contexte et
                                                                                                        typedef struct Gauss_t{
                                                                                                                                                        .globl main
                                                                                        passage de
paramètres par la
passage de
paramètres par la
                                                                                                            int re ;
                                                                                                                                             main:
                 typedef struct Gauss_t{
                                                           .globl main
    pile
                                                                                            pile
                                                                                                            int im ;
                                                                                                                                                   pushl %ebp
                     int re :
                                                     main:
                                                                                                          } Gauss_t ;
                                                                                                                                                   movl %esp, %ebp
                     int im ;
                                                           pushl %ebp
                                                                                                                                                   subl $8, %esp
                  } Gauss_t ;
                                                           movl %esp, %ebp
                                                                                                                                                   andl $-16, %esp
                                                           subl $8, %esp
                                                                                                                                                   movl $1,-8(%ebp)
                                                           andl $-16, %esp
                                                                                                                                                   movl $1,-4(%ebp)
                                                           movl $1,-8(%ebp)
                                                                                                                                                   subl $8, %esp
                                                           movl $1,-4(%ebp)
                                                                                                       int main(void){
                                                                                                                                                   pushl -4(%ebp)
                                                           subl $8, %esp
                                                                                                                                                   pushl -8(%ebp)
                 int main(void){
                                                                                                           struct Gauss_t var ;
                                                                                                                                                   call UN
                                                                                                                                                   addl $16,
                                                                                                           var.re = 1;
                    struct Gauss_t var ;
                                                                                                           var.im = 1;
                                                                                                                                                   movl $0, %eax
                    var.re = 1 ;
                                                                                                                                                   leave
                    var.im = 1 ;
                                                           movl $0, %eax
                                                                                                           UN(var);
                                                                                                                                                   ret
                                                           leave
                                                           ret
                                                                                                           return 0 ;
Passage de
                                                                                       Passage de
                    return 0 ;
                                                        4 D > 4 @ > 4 E > 4 E > E 994 P
                                                                                                                                               4 D F 4 D F 4 E F 4 E F 9 Q C
V74 (09-03-2012)
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf V74 (09-03-2012)
                                                                                                                                               www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf
                                                                                        Pratique du C
Classes
d'allocation,
 Pratique du C
                                                                                                       Fonction retournant une structure
 Classes
d'allocation,
  contexte et
                                                                                         contexte et
                                                                                                        typedef struct Gauss_t{
                                                                                                                                                 .text
passage de
paramètres par la
                                                                                        passage de
paramètres par la
                                                                                                            int re ;
                                                                                                                                                 .globl main
    pile
                                                                                           pile
                 typedef struct Gauss_t{
                                                                                                            int im ;
                                                            .text
                                                                                                                                              main:
                                                     .globl UN
                     int re ;
                                                                                                          } Gauss_t ;
                                                                                                                                                pushl %ebp
                     int im ;
                                                UN:
                                                                                                                                                movl %esp, %ebp
                  } Gauss_t ;
                                                                                                                                                subl $24, %esp
                                                     pushl %ebp
                                                    movl %esp, %ebp
                                                                                                                                                andl $-16, %esp
                 void UN(Gauss_t par){
                                                                                                                                                movl $1, -8(%ebp)
                                                     subl $8, %esp
                                                     movl 8(%ebp), %eax
                                                                                                                                                movl $1, -4(%ebp)
                    par.re = 2 ;
                                                     movl 12(%ebp), %edx
                                                                                                                                                leal -16(%ebp), %eax
                                                     movl %eax, -8(%ebp)
                                                                                                       int main(void){
                                                                                                                                                subl $4, %esp
                                                    movl %edx, -4(%ebp)
                                                                                                                                                pushl -4(%ebp)
                                                     movl $2, -8(%ebp)
                                                                                                           struct Gauss_t var,res ;
                                                                                                                                                pushl -8(%ebp)
                                                     leave
                                                                                                           var.re = 1;
                                                                                                                                                pushl %eax
                                                     ret
                                                                                                           var.im = 1;
                                                                                                                                                call UN
                                                                                                                                                addl $12, %esp
                                                                                                           res = UN(var) ;
                                                                                                                                                movl -16(%ebp), %eax
                                                                                                           var.im = res.re ;
                                                                                                                                                movl %eax,-4(%ebp)
Passage de
                                                                                        Passage de
                                                                                                           return 0 ;
                                                                                                                                                movl $0,%eax
                                                                                                                                                leave ret
                                                                                                                                               4 D F 4 B F 4 E F 4 E F 9 Q C
                                                        40 ) 40 ) 42 ) 42 ) 2 990
V74 (09-03-2012)
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf V74 (09-03-2012)
                                                                                                                                               www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf
```

```
Pratique du C
Classes
d'allocation,
contexte et
passage de
paramètres par la
pile
                 typedef struct Gauss_t{
                                                                  .text
                     int re ;
                                                             .globl UN
                     int im ;
                   } Gauss_t ;
                                                          pushl %ebp
                                                          movl %esp, %ebp
                 struct Gauss_t UN(Gauss_t par){ subl $8, %esp
                                                          movl 8(%ebp), %eax
                    par.re = 2 ;
                    return par ;
                                                          movl 12(%ebp), %edx
                                                          movl 16(%ebp), %ecx
                                                          movl %edx, -8(%ebp)
                                                          movl %ecx, -4(%ebp)
                                                          movl $2, -8(%ebp)
                                                          movl -8(%ebp), %edx
                                                          movl -4(%ebp), %ecx
                                                          movl %edx, (%eax)
                                                          movl %ecx, 4(%eax)
                                                          leave
                                                          ret $4
Passage de 
paramètres
                                                         www.fil.univ-lille1.fr/~sedoglav/C/Cours09.pdf
V74 (09-03-2012)
```