# Du C à l'objet

#### main.c

```
int abc = 1;  // --> Données RW
char *str; // --> BSS
const int ri = 10; // --> Données R only
main() {
 int a=1,b=2,c; // ---> variables locales : pile
  char *ptr; // ---> variable locale : pile
  ptr = malloc(4); // ----> Tas
  c= a+b; // ----> Texte
```

#### main.o

Entête
30 : call malloc 38 : add
abc=1 const int i =10
malloc : 30
main abc ri
malloc
30 : main.c:7

Text

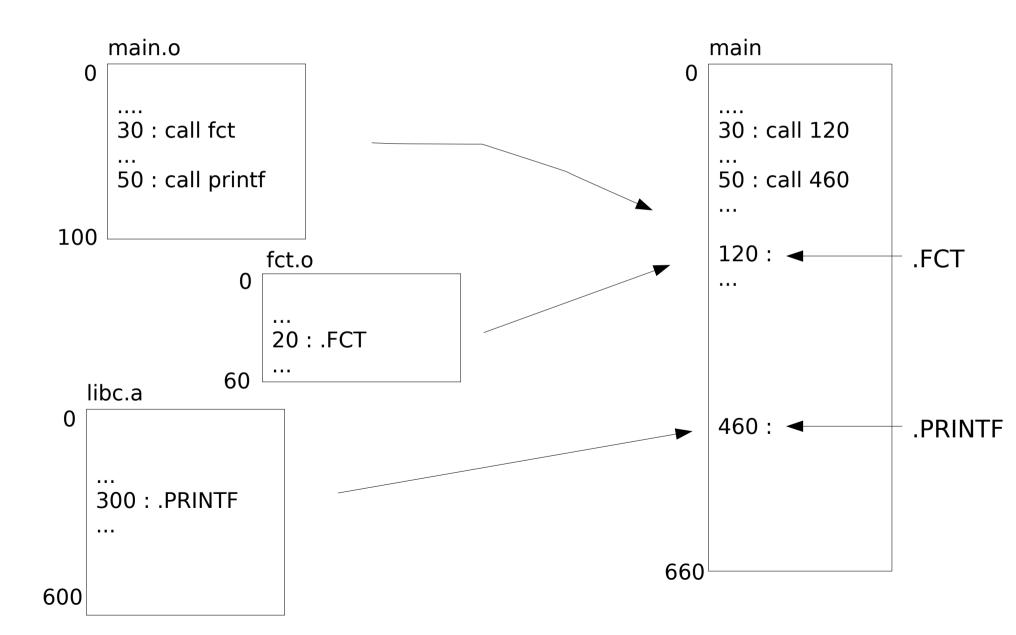
Data

Relocation

Symbol table

Debug info

### De l'objet à l'exécutable



# De l'exécutable au processus

Chargeur

Exécutable (fichier sur disque)

Entête

Instructions

Données

Table des symboles

Info de mise au point

Processus en RAM

Adresse 0

Instructions

Données

**BSS** 

Tas



Pile

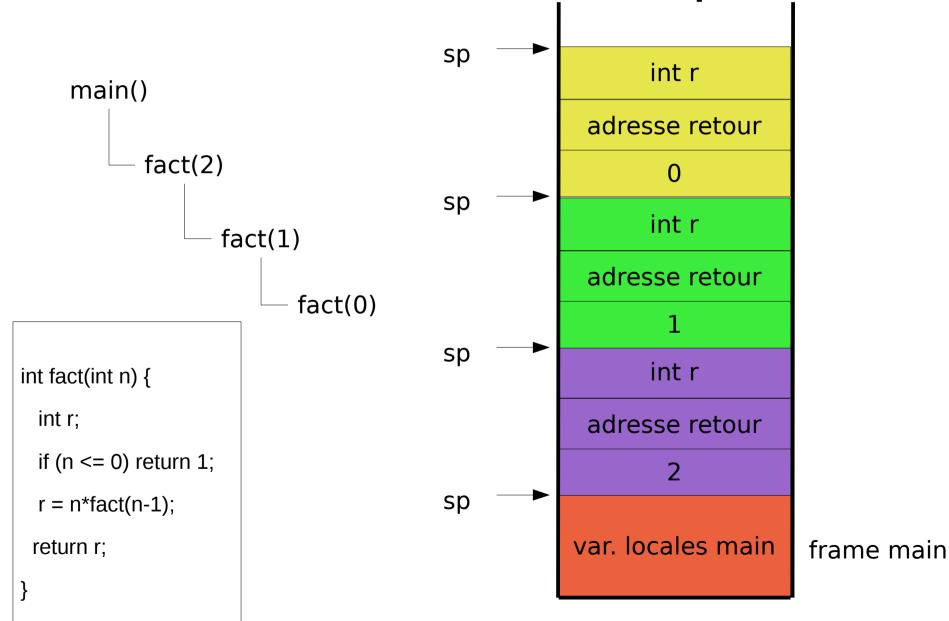
Adresse FF..FF

# Allocation dynamique

```
Instructions
main () {
                                                                      Données
                                                                        BSS
int *ptr1, *ptr2;
                                           adresse 1000
                                                               18
ptr1 = (int *)malloc(sizeof(int)*4);
ptr2 = (int *)malloc(sizeof(int)*8);
                                                                      segmentation fault
ptr1[1] = 18;
ptr1[30] = 15;
                                                                        Pile
                                                                              1016
                                                                                     1000
                                                                               ptr2
                                                                                      ptr1
```

```
int fact(int n) {
    int r;
    int x;
    if (n <= 0) return 1;
    r = n*fact(n-1);
    return r;
}

main() {
    int x;
    int x;
    ...
    x = fact(2);
    return r;
}</pre>
```



```
main()
             fact(2)
                      fact(1)
                               fact(0)
int fact(int n) {
  int r;
  if (n <= 0) return 1;
  r = n*fact(n-1);
 return r;
```

int r adresse retour int r adresse retour int r adresse retour var. locales main

```
main()

fact(2)

fact(1)
```

```
int fact(int n) {
  int r;
  if (n <= 0) return 1;
  r = n*fact(n-1);
  return r;
}</pre>
```

int r adresse retour int r adresse retour var. locales main

```
main()

fact(2)

fact(1)
```

```
int fact(int n) {
  int r;
  if (n <= 0) return 1;
  r = n*fact(n-1);
  return r;
}</pre>
```

int r adresse retour int r adresse retour var. locales main

```
int r
     main()
                                                         adresse retour
                                                                 0
                                                               int r
                                                         adresse retour
                                                                 1
                                                               int r
int fact(int n) {
 int r;
                                                         adresse retour
                                          sp
 if (n <= 0) return 1;
 r = n*fact(n-1);
                                                       var. locales main
 return r;
```