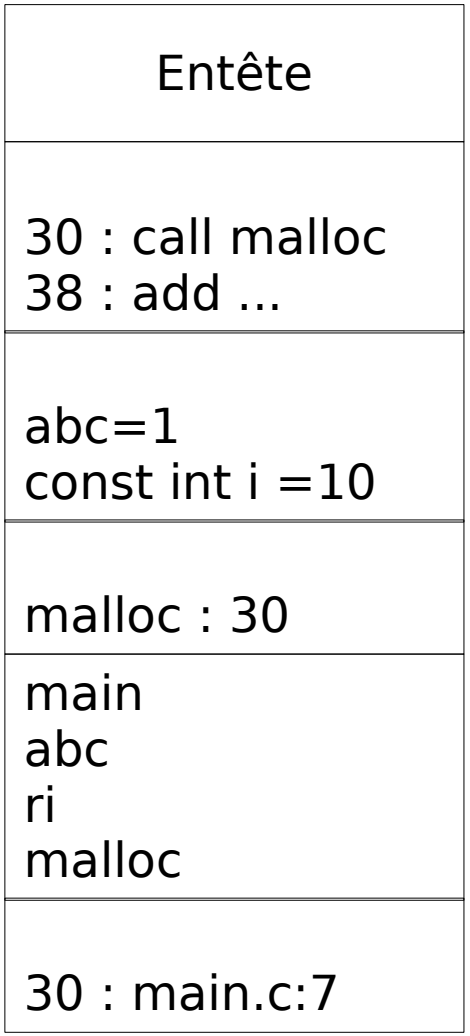
Du C à l'objet

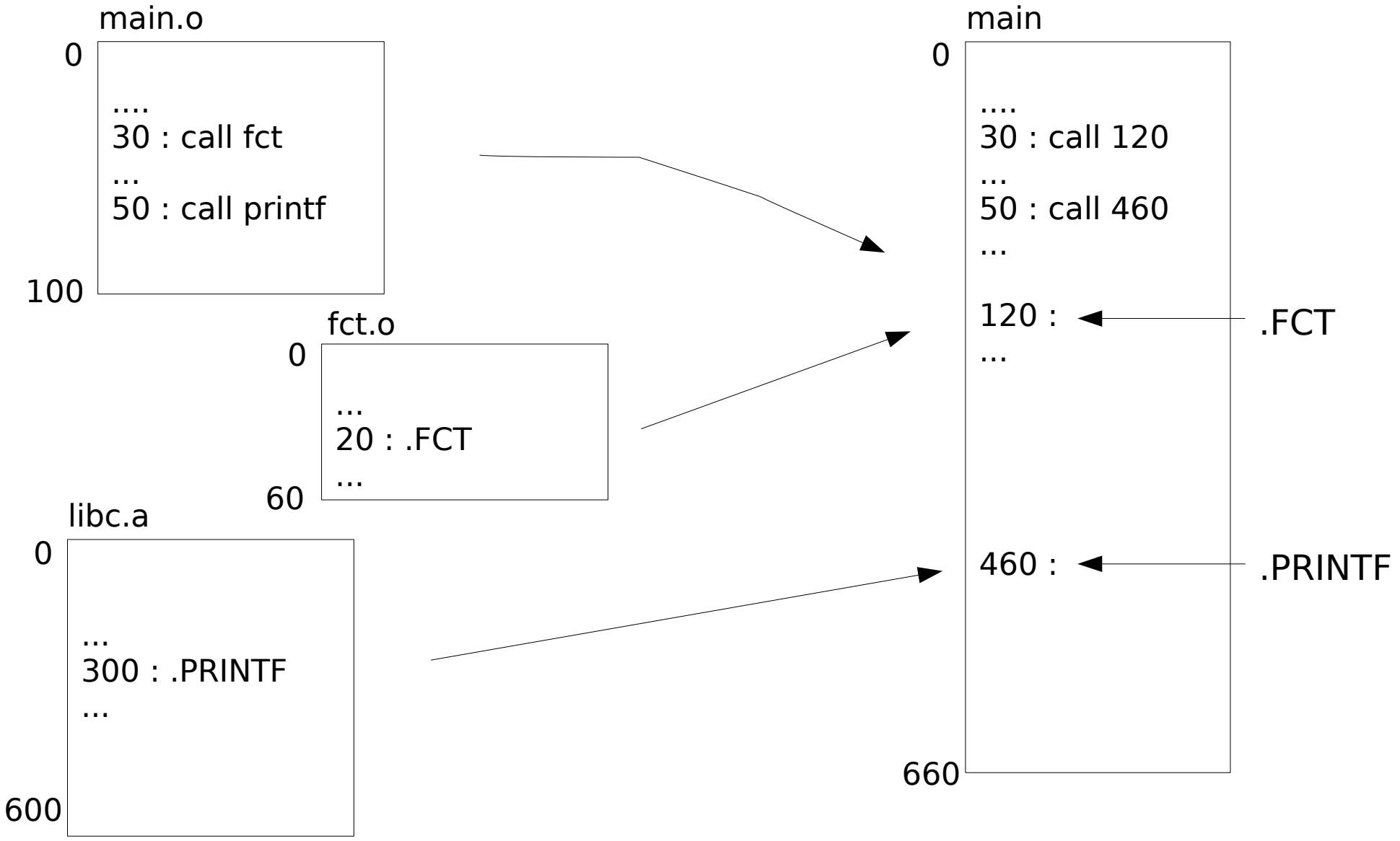
main.c

```
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
}
```

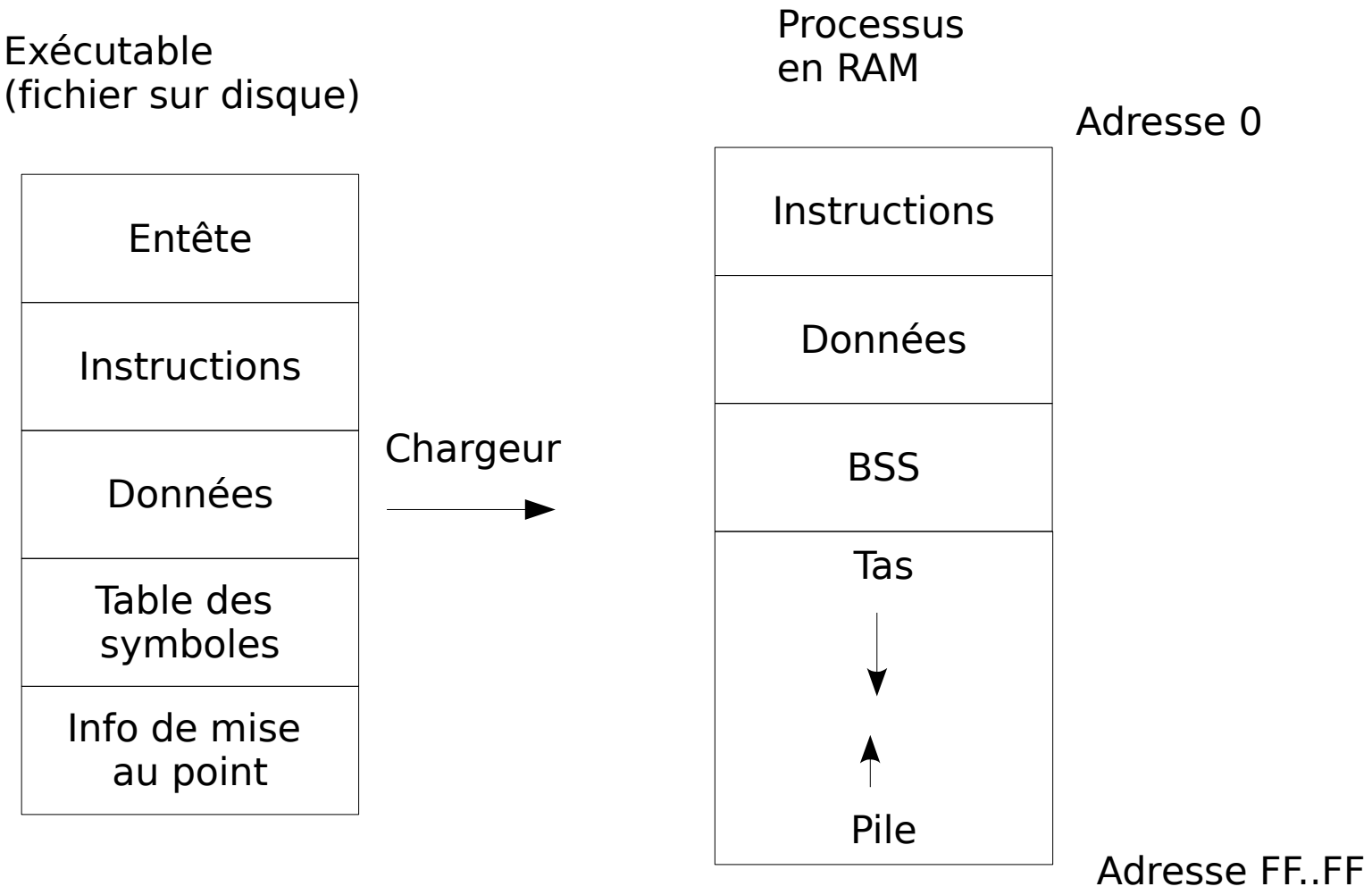
main.o



De l'objet à l'exécutable



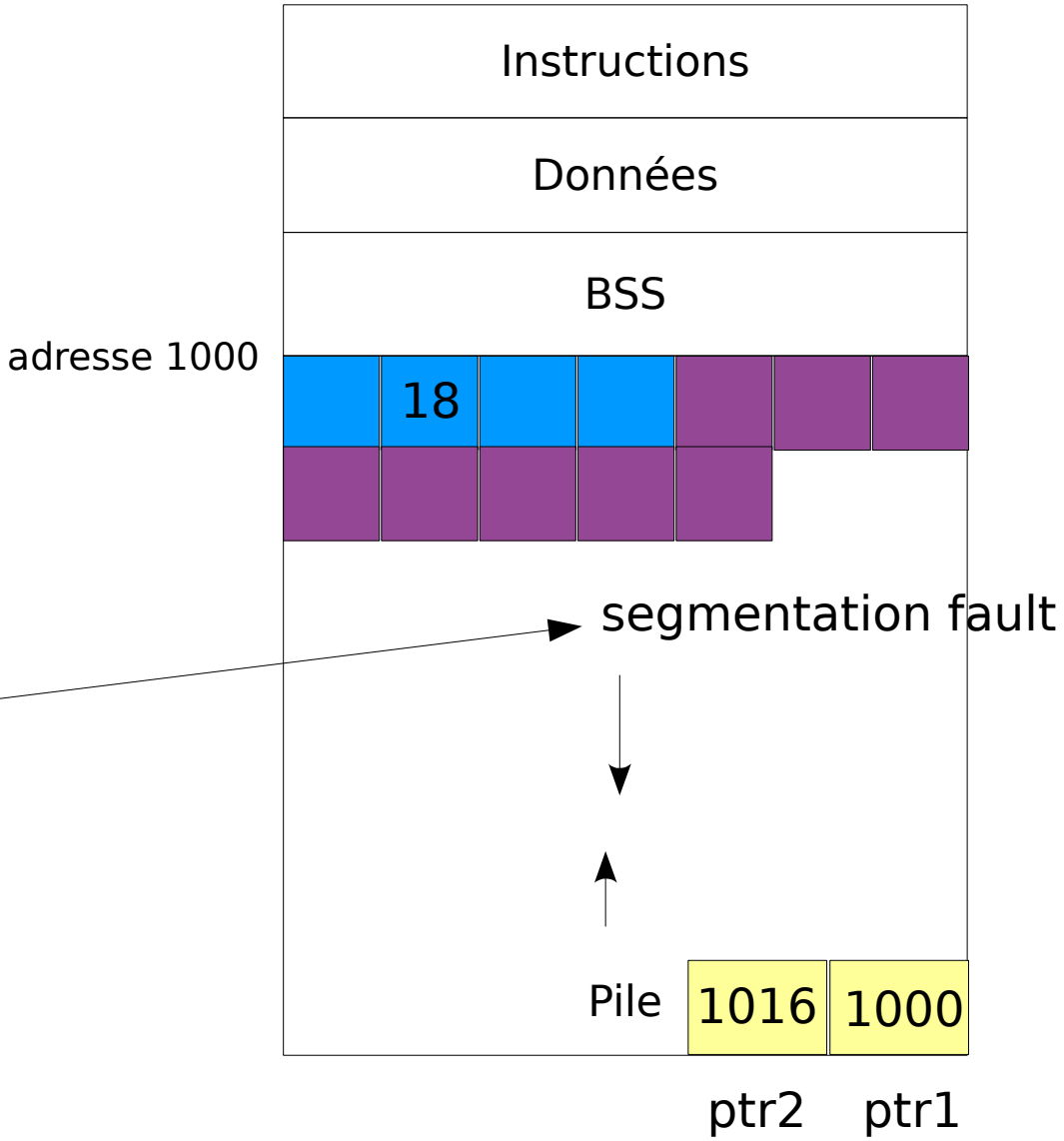
De l'exécutable au processus



Allocation dynamique

```
main () {
…
int *ptr1, *ptr2;

ptr1 = (int *)malloc(sizeof(int)*4);
ptr2 = (int *)malloc(sizeof(int)*8);
ptr1[1] = 18;
ptr1[30] = 15;
…
}
```



Fonctionnement de la pile

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

```
main() {
 int x;
  ...
  x = fact(2);
  ...
}
```

Fonctionnement de la pile

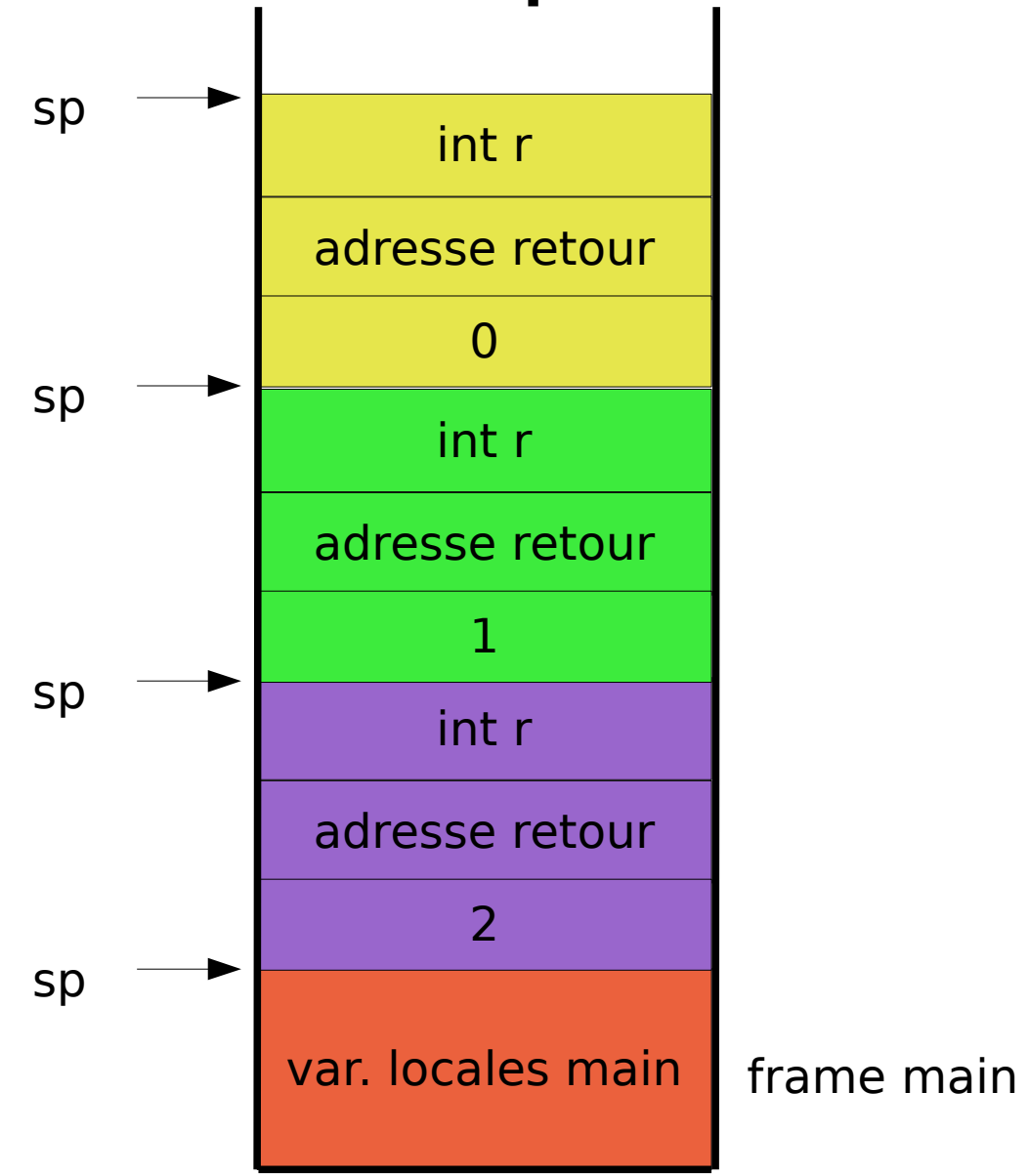
main()

fact(2)

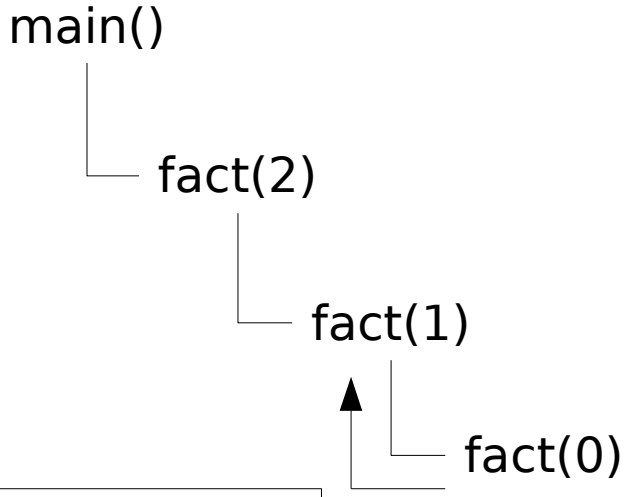
fact(1)

fact(0)

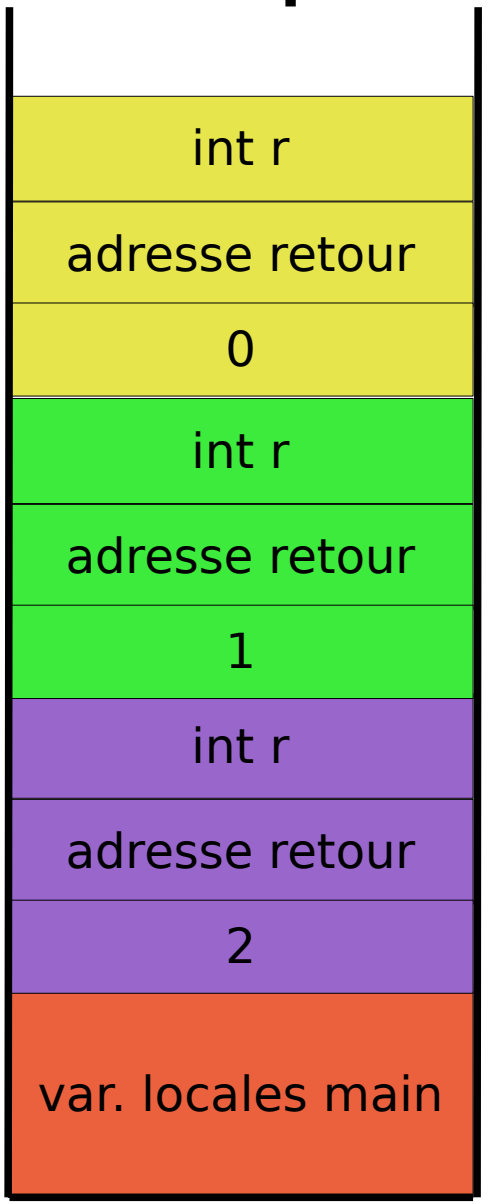
```
int fact(int n) {
   int r;
   if (n <= 0) return 1;
   r = n*fact(n-1);
  return r;
}
```



Fonctionnement de la pile




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



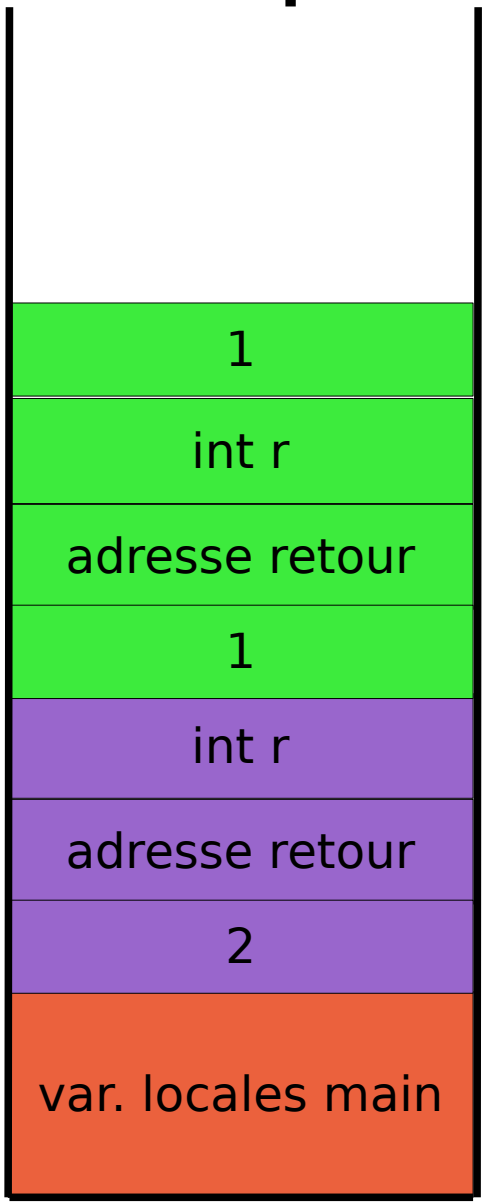
Fonctionnement de la pile

main()

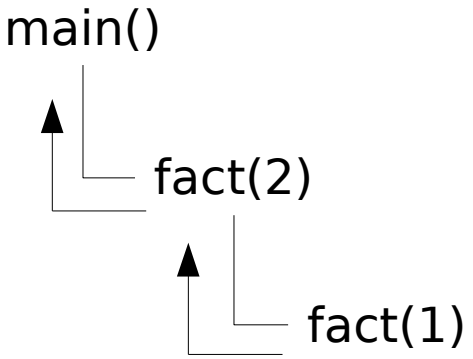
fact(2)

fact(1)

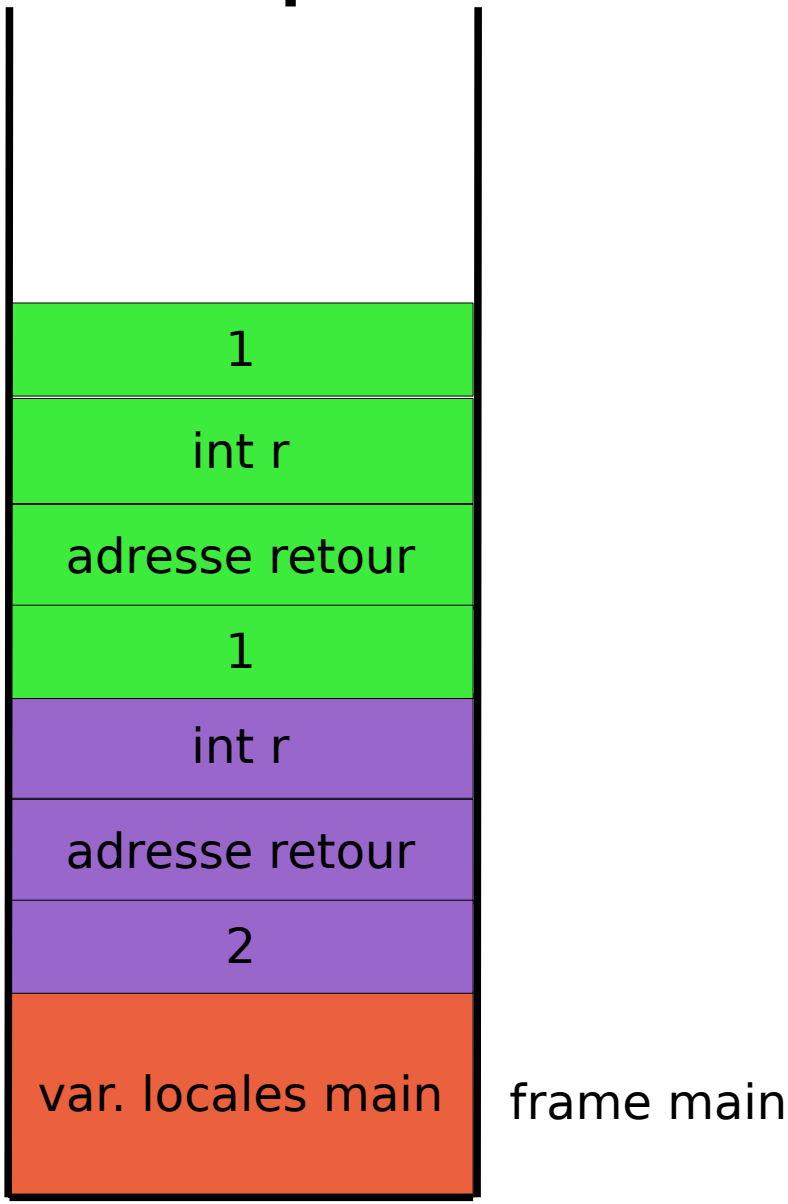
```
int fact(int n) {
   int r;
   if (n <= 0) return 1;
   r = n*fact(n-1);
  return r;
}
```



Fonctionnement de la pile




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



Fonctionnement de la pile

main()

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