

C <pre>int *p = malloc(1);</pre>	Clight <pre>(* declas: Etempvar tmp (void *\ ) Evar p (int *\ ) *) Sseq (Sbuiltin (Some tmp) malloc [1])       (Sassign (Evar p) (Etempvar tmp))</pre>	CFML <pre>(* before pre-proc *) trm_let p heap (val_alloc 1)  (* after pre-proc *) trm_let tmp const (val_alloc 1)       (trm_let p heap tmp &lt;cont&gt;)</pre>
<pre>*p = v;</pre>	<pre>Sassign (Ederef (Evar p)) v</pre>	<pre>(val_set p v)</pre>
<pre>*p;</pre>	<pre>Ederef (Evar p)</pre>	<pre>(val_get p)</pre>
<pre>p;</pre>	<pre>Evar p</pre>	<pre>p</pre>
<pre>free p;</pre>	<pre>Sbuiltin None free (Evar p)</pre>	<pre>(val_free p)</pre>
<pre>int x;</pre>	En vrai difficile à voir comme cas <pre>(* decla: Evar x (int) *)</pre>	<pre>trm_let x stack (val_alloc 1)</pre>
<pre>x = v;</pre>	<pre>Sassign (Evar x) v</pre>	<pre>(val_set x v)</pre>
<pre>x;</pre>	<pre>Evar x</pre>	<pre>(val_get x)</pre>
<pre>&amp; x;</pre>	<pre>Eaddrof (Evar x)</pre>	<pre>x</pre>
<pre>register int x = 3;</pre>	<pre>Sset x (Eint 3)</pre>	<pre>trm_let x const (val_int 3)</pre>
<pre>x;</pre>	<pre>Etempvar x</pre>	<pre>(val_get x)</pre>
<pre>f(x); (si résultat utile)</pre>	<pre>(* decla: Etempvar tmp (void *\ ) *) Scall Some tmp f [x]</pre>	<pre>trm_let y const f [x]</pre>
<pre>f(x); (sinon)</pre>	<pre>Scall None f [x]</pre>	<pre>trm_let bind_anon const f [x]</pre>
<pre>if (e) then s1 else s2</pre>	<pre>Sifthenelse e s1 s2</pre>	<pre>(trm_ite e s1 s2)</pre>
<pre>while (e) s (voir comment on compile si la cond a des side-effects (same pour ite))</pre>	<pre>Swhile e s</pre>	<pre>trm_while e s</pre>

---

## Compilation phases:

`while t1 do t2`

```
while (1) {  
     $\llbracket t1 \rrbracket_b$ ;  
    if (!b) {  
        break;  
    }  
    t2;  
}
```

Tout appel de fonction ->

```
let y = f x in
```

(entraîne la décl de y comme tempvar)

---