

```
struct huge {
    char member[4096];
};
```

```
extern void f1(struct huge s);
extern void f2(struct huge *s);
```

```
struct huge glbl;
```

```
void f(void) {
    f2(&glbl);
}
```

↳ indirizza

```
pushq    %rbp
movq     %rsp, %rbp
movq     $glbl, %rdi
call     f2
popq     %rbp
ret
```

↳ PUNTATORE a STRUTTURA:

- In C, il passaggio di parametri è sempre fatto *per valore*
  - La funzione riceve *una copia* del parametro
    - in caso di vettori/matrici, *il passaggio per valore riguarda il puntatore associato!*
  - Il motivo è evitare *side effect* nascosti nell'invocazione di funzioni (anche di libreria)

```
struct huge {
    char member[4096];
};
```

```
extern f1(struct huge s);
extern f2(struct huge *s);
```

```
struct huge glbl;
```

```
void f(void) {
    f1(glbl);
}
```

```
pushq    %rbp
movq     %rsp, %rbp
subq     $4096, %rsp
movq     %rsp, %rdi
movq     $glbl, %rsi
movq     $512, %rcx
cld      → SE df=0, la cop'ia viene fatta in avanti!
rep movsq
call     f1
addq     $4096, %rsp
leave
ret
```