No Comment – come e perché scrivere i commenti nel C (e C++)

Dom Cobb: *Ég hef allt undir stjórn.*
Arthur: *hið gagnstæða væri alvarlegt.*

(*…una premessa: questo post è un remake* ***di un mio vecchio post****. Ma, anche se tratta lo stesso argomento, amplia un po' il discorso è mi è sembrato il caso di riproporlo. Leggete e mi direte…*)

Non so se avete visto lo splendido **Inception** di **Christopher Nolan**. Bene, provate a pensare come sarebbe vedere un film con una trama così complessa doppiato in una lingua che non conoscete, come l'**islandese** per esempio (*e senza sottotitoli!*). Ci capireste qualcosa? Cosa si dicono Dom Cobb e Arthur qua sotto? *Ecco, i sottotitoli sono i commenti del Cinema…*



…adesso ti spiego: Ég hef allt undir stjórn…
Se siete di quelli che *“Non scrivo commenti. Sono una perdita di tempo”* fermatevi qui. Tanto non riuscirei a convincervi (*e neanche ci tengo*).

Per tutti gli altri: il commento è un vostro amico fedele, non vi abbandonerà mai. Il commento è l’ultima barriera tra il capire e il non capire *“Cosa avevo in testa quando ho scritto quel codice un anno fa ?”*. Il commento è la vostra dimostrazione di rispetto per i colleghi di lavoro: prima o poi, qualcuno di loro metterà le mani sul vostro *Software, e il numero di maledizioni che vi manderà sarà inversamente proporzionale al numero (e alla qualità) dei commenti che avete scritto.*

Certo, ci sono varie maniere di commentare il codice: si va dal “**Absolute No Comment**” al “**Auto-Commentante con aggiunta di commenti**” (*beh, quest’ultimo è un po’ esagerato*), con tutte le sfumature intermedie possibili…

Facciamo una piccola dimostrazione: esaminiamo tre maniere si scrivere una funzione di lettura di un dispositivo di controllo industriale: si tratta, nel **Caso 3**, di codice quasi reale (*solo quasi, eh!*) che ho scritto alcuni anni fa: per questo esempio l’ho solo adattato un po’. Si noti che i tre codici sono praticamente identici, anche se la lettura dà tutt’altra impressione…

Caso 1: codice Absolute No Comment

```
int smsg(Dev *p)
{       int       n=0;       if       (p->msg[0])       {
snprintf(p->btx,sizeof(p->btx),"data:   %s",p->msg);   if
(!(n=(sdata(p->ch,(char*)p->btx,strlen(p->btx))))) return -1;
p->msg[0]=0; } return n;
}
```

Come avrete notato non ci sono commenti, il codice è (a dir poco) **criptico**, e si risparmia su tutto: nomi, spazi, linee: *non si sa mai che qualcuno riesca perfino a capire cosa c’è*

scritto. **Per maniaci della segretezza**.

Caso 2: codice auto-commentante

```
int sendMessageToDeviceXYZ( DeviceXYZ *p_deviceXYZ)
{ int number_sent = 0; if (p_deviceXYZ->message_to_send[0]) {
snprintf(p_deviceXYZ->buffer_tx,
sizeof(p_deviceXYZ->buffer_tx),     "data:     %s",
p_deviceXYZ->message_to_send);  if  ((number_sent  =
sendDeviceData(p_deviceXYZ->channel_tx,       (char
*)p_deviceXYZ->buffer_tx, strlen(p_deviceXYZ->buffer_tx))) ==
-1) { return -1; } p_deviceXYZ->message_to_send[0] = 0; }
return number_sent;
}
```

Come avrete notato non ci sono commenti, ma il codice è **auto-esplicativo** e non si va al risparmio. I nomi sono scelti per la massima chiarezza. **Per chi non ama i commenti ma vuole essere chiaro a tutti i costi.**

Caso 3: codice chiaro e commentato

```
/* NAME * sendMsgDev - invia un messaggio al dispositivo *
SYNOPSIS * int sendMsgDev( * DevXYZ *dev); // dispositivo
destinazione * DESCRIPTION * sendMsgDev() invia un messaggio
al dispositivo di tipo XYZ. L'argomento <dev> * è un pointer
alla struttura dati del dispositivo. * RETURN VALUE * In caso
di successo, sendMsgDev() restituirà il numero di caratteri
inviati. * Altrimenti, verrà restituito -1. */ int sendMsgDev(
DevXYZ *dev) // dispositivo destinazione
{ // reset numero di char spediti int n_sent = 0; // controllo
se c'è un messaggio da spedire if (dev->msg_2_snd[0]) { //
formatto il messaggio nel buffer di trasmissione
snprintf(dev->buf_tx, sizeof(dev->buf_tx), "data: %s",
dev->msg_2_snd); // spedisco il messaggio al dispositivo if
((n_sent = sendData(dev->chan_tx, (char *)dev->buf_tx,
strlen(dev->buf_tx))) == -1) { // errore di invio return -1; }
// messaggio spedito: reset del contenuto dev->msg_2_snd[0] =
0; } // ritorno il numero di char spediti return n_sent;
}
```

Come avrete notato ci sono una intestazione (abbastanza completa, ma si può espandere) e molti brevi commenti. Il

codice non è auto-esplicativo ma è sufficientemente chiaro. **Per comment-lovers**.

Qualcuno potrebbe dire che in questo ultimo caso ci sono fin troppi commenti, e magari molto ovvi: ma io penso che *Melius abundare quam deficere* e se a qualcuno i commenti non interessano **può anche non leggerli!** O dà così fastidio che ci siano dei dettagli che solo alcuni possano trovare utili? Bisogna ricordare che l'obiettivo del **Caso 3** è che si possa interpretare il codice solo leggendo i commenti, **ma non è obbligatorio farlo**! Ci sono troppi commenti? **E allora non leggerli e non rompermi i cabasisi**!

Inutile dirlo: io appartengo alla scuola **comment-lovers**. Però rispetto anche i seguaci del codice **auto-commentante**. Sui signori del **Absolute No Comment** l'unica cosa che posso dire è… **No Comment**. Comunque, se vi può interessare, il **Caso 1** non è una forzatura, *e ho visto anche di peggio, non fatemi parlare…* Anzi, si, parlerò! E vi racconterò tre aneddoti:

- Ho sentito con le mie orecchie (*true story*) frasi di questo tipo: *"Ho dovuto fare dei cambi a quel Software che avevi scritto tu: inutile dirti che, grazie al codice chiarissimo e stra-commentato, ho potuto farlo in un attimo. Grazie!"* (il *"tu"* della frase è uno specialista del **Codice Chiaro e Commentato**, *non faccio nomi, ah ah ah*).
- Ho sentito con le mie orecchie (*true story*) frasi di questo tipo: *"Devo mettere le mani sul Software scritto da quel mascalzone di Tizio Caio… Ci metterò una settimana solo per capire dove fare le modifiche! Per fortuna se ne è andato, ma ci ha lasciato una bella eredità…"* (il sig.*"Tizio Caio"* è uno specialista del codice **Absolute No Comment**, *ometto il nome vero…*).
- Ho lavorato in posti dove ho visto (*giuro!*) *file* di migliaia di linee **non auto-commentanti** e senza neanche un commento (*orrore! orrore!*).

Ecco, so che quanto sto per dire vi sembrerà un po’ radicale (*e in effetti lo è*), ma lo dirò lo stesso: se quanto ho scritto nell’articolo vi da fastidio o vi sembra inutile può essere dovuto solo a due motivi:

1. Sicuramente non avete mai lavorato su *Software* scritto da altri… *siete proprio fortunati!*
2. Siete quel mascalzone di Tizio Caio (*o siete uguali a lui*).

Ok, per oggi può bastare. Chiedo scusa se ho offeso qualcuno ma, tutto sommato, *chi si è offeso ha la coda di paglia e se lo merita, ah ah ah.*

Ciao e al prossimo post!

https://italiancoders.it/no-comment-come-e-perche-scrivere-i-commenti-nel-c-e-c/