

## Raccomandazioni

Seguono alcune considerazioni sugli errori riscontrati più di frequente tra gli elaborati dei partecipanti, nonché delle raccomandazioni per un buon esito delle prove al calcolatore:

- dato un buffer `buf`, `sizeof(buf)` non restituirà il numero di byte presenti (i.e., scritti) attualmente nel buffer, bensì il numero di byte occupati da un puntatore (4 su IA32, 8 su x86\_64) se allocato dinamicamente, o il numero di byte per esso allocati staticamente; si noti che nel caso in cui `buf` contenga una stringa NULL-terminated, è possibile utilizzare `strlen(buf)` per conoscerne l'effettiva lunghezza
- quando si deve invocare una funzione per la programmazione di sistema:
  - fare attenzione se i singoli parametri siano o meno dei puntatori
  - verificare se può essere soggetta ad interruzioni, e in tal caso gestirle
  - verificare il comportamento della funzione in caso di errori: inserire codice idoneo a trattare errori gestibili, utilizzare la macro opportuna per gli altri
  - per funzioni della libreria `pthread`, si vedano il capitolo 2 e l'appendice A.2 della dispensa di UNIX system programming
- quando viene acquisita una risorsa (socket aperta, area di memoria allocata, etc.), questa deve poi essere rilasciata in maniera opportuna
- utilizzare puntatori non inizializzati porta ragionevolmente ad un Segmentation fault!
- i cast di puntatori da e verso `void*` sono impliciti in C, renderli espliciti o meno è una scelta di natura puramente stilistica nella programmazione
- fare molta attenzione nell'aggiornamento dell'indice (o del puntatore) con cui si opera su un buffer: disallineamenti  $\pm 1$  sono tra le cause più comuni di errori
- quando si vuole passare un puntatore ad una funzione, aggiungere al nome della funzione una coppia di parentesi tonde equivale invece ad invocarla!
- **i commenti nel codice contengono molte informazioni utili per lo svolgimento della prova, si consiglia quindi di tenerli in debita considerazione**