

RELAZIONE PROGETTO

Giorgio Paoloni, 1883570.

COSA È BOT:

“Bot” è un programma scritto in linguaggio C che ha come scopo riprodurre alcune delle funzionalità del programma “Top” (o simili quali “htop” ecc...) disponibile per sistemi GNU/Linux.

Quindi, citandone alcune: mandare dei segnali ai processi (kill, sleep,...), visionare i processi, ordinare i processi, visualizzare delle statistiche generiche dell'utilizzo risorse del sistema,...

Note: Bot è compatibile con sistemi operativi che adottano syscall e pseudo-fs di Linux. Non funziona su sistemi operativi come Windows o MacOS. Il programma è stato testato su EndeavourOS (Arch-based) e Ubuntu (Debian-based).

OBIETTIVI DEL PROGRAMMA:

I due obiettivi principali del programma sono:

- 1) Ottenere le informazioni dal sistema operativo tramite delle chiamate a sistema, utilizzo di cartelle nello pseudo-filesystem (/proc), ecc...
Queste informazioni ottenute verranno interpretate, rese più comprensibili e se necessario stampate a schermo.
- 2) Sfruttare la libreria “NCurses” per una TUI, Text-based UI, a scopo di fornire all'utente una parte grafica semplice ma efficace con cui interagire. Si è cercato il più possibile di rendere il programma responsivo. La grafica è ben divisa e coerente, inoltre è molto intuitiva.

NOTE DI PROGRAMMAZIONE:

Si è cercato di creare delle funzioni richiamabili, per aumentare la modularità e la leggibilità del codice.
Sfruttare il più possibile librerie e funzioni standard C.
Evitare allocazioni dinamiche (heap) ove non strettamente necessarie.
Evitare di allocare strutture inutilmente.
Differenziare la parte di presentazione (TUI.c) da quella funzionale (proc.c e process_monitor.c).

Questo ha permesso aggiunte e modifiche semplici e veloci. Un programma intuitivo e reattivo.

METRICHE UTILIZZATE:

Tutte le informazioni sono prese da `/proc` e sotto-elementi. Sono anche sfruttate chiamate a sistema es: per ottenere la frequenza del processore. Su internet o terminale la documentazione di `/proc` è facilmente accessibile attraverso `"man /proc"`, però alcune metriche vengono segnalate dal manuale come non del tutto precise oppure possono esserci errori di approssimazione dovuti a casting o troncamenti della divisione.

Inoltre anche se `/proc` è ben documentato. L'utilizzo di questi valori per ottenere metriche "derivate" non lo è.

Ho cercato di evitare il più possibile di allocare strutture per salvare le metriche fornite da `/proc`, poiché ho reputato il salvataggio ridondante e rallentante, essendo già presente nel pseudo-filesystem. E sfruttarle a run-time. Questo permette di avere delle informazioni il più possibile recenti e aggiornate. Oltre che un risparmio di memoria.

Alcune strutture sono necessarie perché servono ad allocare metriche "derivate" non presenti in `/proc` né facilmente calcolabili.

Le strutture sono principalmente allocate staticamente (stack) e sono sfruttate per tutta l'esecuzione del programma.

Note: Con metriche "derivate" intendo metriche non disponibili direttamente su `/proc` ma ottenibili attraverso calcoli algebrici con altre metriche e/o chiamate a sistema (syscall).

UTILIZZO DI NCURSES:

NCurses è stata fondamentale per fornire, in maniera standard e portabile, un rudimento di grafica testuale tramite cui l'utente può interagire col programma. Le funzioni della libreria utilizzate sono quelle necessarie per fornire una grafica semplice ma potente: funzioni di print, refresh, get,... si noteranno nel codice sorgente del programma.

L'obiettivo non è stato quello di sfruttare a pieno la libreria, ma piuttosto utilizzare solo le funzioni necessarie.

TEST VALGRIND:

Il programma è stato testato tramite il tool Valgrind per testare memory leak di tipo lost: strutture non deallocate, strutture riallocate con puntatori persi, file aperti ma non chiusi,...

GUIDE SEGUITE:

Il codice sorgente è ben commentato, praticamente ogni scelta o guida seguita è spiegata.

Sono stati ampiamente anche usati siti come stackoverflow,...

Se qualche snippet di codice è stato copiato, per velocizzare la scrittura del codice, è stato modificato per lo scopo, analizzato e citata la provenienza dello snippet.

Le guide principalmente seguite sono:

-proc: <https://man7.org/linux/man-pages/man5/proc.5.html>

-ncurses: <https://invisible-island.net/ncurses/howto/NCURSES-Programming-HOWTO.html>

-signal: <https://man7.org/linux/man-pages/man7/signal.7.html>

-pthread: <https://man7.org/linux/man-pages/man7/pthreads.7.html>

-valgrind: <https://valgrind.org/docs/>

