

Két új szkript az OpenVPN-kapcsolat időszakos ki- és bekapcsolására

PCLinuxOS Magazine - 2026. június

Írta: Paul Arnote (parnote)

Már biztosan említettem, hogy az internetes ügyleteim során hajlok arra, hogy VPN-t használjak. Napjaink kíváncsiskodó társadalmában a VPN-t a személyes adataim védelméért folytatott küzdelem élharcosának tekintem ... nos ... privát.



Gino Crescoli képe a Pixabay-ről

Szerintem, semmi ésszerű ok nincs arra, hogy az internetszolgáltató figyelje, vagy lássa, mit csinálok az interneten. Nem arról van szó, hogy bármi illegálisat csinálnék. A szolgáltató azonban két fő okból szeretné elérni online adataimat. Először is, a „dadám” akar lenni, hogy biztos ne csináljak semmi illegálisat. A szabályok és a törvények betartását céloznak tartom, amik közül néhány annyira megterhelő, hogy szinte lehetetlen maradéktalanul betartani. Tehát nem én vagyok az a „valaki”, akit keres. Másodsor, az internetszolgáltató az érdeklődési körömhöz akarja igazítani a reklámokat. Köszönöm, nem kérem! Eleget kell nyelnem a többi szolgáltató miatt, ezért öröm és megkönnyebbülés, ha az internetszolgáltatómat kizárhatom a buzgólkodók csoportjából.

A VPN-nel az szolgáltató nem lehet a „dadám”. A számítógépem és a VPN-szolgáltató szerverei között a teljes forgalom titkosított, így csak a forgalmat látják. Amit nem látnak az az, hogy hol voltam, mit csináltam ott, vagy bármi mást... csak a forgalmat. SOK internetszolgáltató azonnal arra gondol, hogy torrent-ezem, akkor biztosan valami illegális dolgot csinálok.

Persze, ez a nagyon „bigott” nézet teljesen figyelmen kívül hagyja a torrent-ek jogos használatát (például a PCLinuxOS legújabb kiadásának letöltését). Természetesen sokan használnak torrent-eket törvények elkerülésére és szerzői joggal védett tartalmak letöltésére az internetről, így engem is „ilyenek” tekintenek. De nem minden torrent vesz részt illegális tevékenységben. Nagyon sokan használnak torrent-eket a FOSS programok letöltésére. A torrent nagyszerű eszköz az INGYENES szoftverek terjesztésére.

Ráadásul, mivel az utóbbi időben a személyes és privát adatok (PPI) a szó szerinti „valutává” váltak, jobban érzem magam, ha tudom, hogy az online továbbított információk előnyükre alkalmazzák a titkosítást a számítógépet elhagyó minden adat esetén. Nem kell ezüst vagy arany tányéron átadni a PPI-t a kreténeknek, ha nem muszáj (és én biztos nem akarom).

Tehát nagyjából ennyi, AMIÉRT VPN-t használok. De vannak bizonyos webhelyek, amik nem hajlandók bekapcsolt VPN-nel kommunikálni. Sok más mellett, az MLB.com és a Netflix is ilyen. Emiatt kerestem egy egyszerűbb módot a VPN ideiglenes letiltására, hogy csatlakozhassam ezekhez a webhelyekhez. És miután végeztem az Ilyen „offenzív” webhely meglátogatásával, ami nem teszi lehetővé a VPN-kapcsolódást, ugyanolyan egyszerű módot akarok a VPN újraaktiválására.

Bármennyire is 1000%-ig, teljesen elégedett vagyok azzal, hogy a PCLinuxOS néhány évvel ezelőtt NetworkManager-re állt áll, a NetworkManager-nek van egy olyan aspektusa, aminek kevésbé örülök. A VPN-menüjének elérése a VPN ki- vagy bekapcsolásához gyakran frusztráló élmény. Elég, ha az egérkurzor akár egy pixelt is kimegy a menü

Két új szkript az OpenVPN-kapcsolat időszakos ki- és bekapcsolására



határain kívülre, és a menü eltűnik, így mindent előlről kell kezdeni. Az akkor válik igazán idegesítővé, amikor a VPN-kiszolgálók nagyon hosszú listájának végén megjelenik végre az a VPN-kiszolgáló, amihez csatlakozni szeretnél.

A VPN ki-, vagy bekapcsolását ez frusztrálóvá teszi. Arra gondoltam, hogy kell lennie egyszerűbb, kevésbé fárasztó módszernek a VPN kikapcsolására (vagy bekapcsolására). Szerencsére a NetworkManager által biztosított parancssori eszközöknek köszönhetően van rá mód.

Mielőtt belevágnánk a VPN-emet ki-, vagy bekapcsoló megoldásom (megoldásaim) megvitatásába, el kell mondanom, hogy ezeknek a szkripteknek a megírása kihívást volt. Úgy éreztem, hogy egy kicsit jobban feszegeti a bash-szkriptelő képességeim határát, mint ami kényelmes. Természetesen, ezzel fejlesztem a képességeimet, és úgy érzem, hogy a bash-szkriptelési készségeimnek használt ez a kirándulás.



VPN váltása (toggle)

Az első szkriptem a „Toggle VPN” (VPN alternálás) Nincs hivatalos „ablaka”. Ehelyett létrehoz egy ikont, ami az értesítési területre kerül. Az ikonhoz jár egy menü, ami az ikonra való jobb egérgombbal kattintva aktiválható.

A kettő közül, ez a szkript a rövidebb. Noha ez egy bash szkript, vannak grafikus felhasználói felületelemei a végfelhasználó számára. Tehát ez a szkript elhelyezhető az asztali környezet „AutoStart” könyvtárában, vagy ott, ahol az automatikus indítási információkat tárolod. Természetesen, letölthető a szkript a [magazin szerveréről](#). A fájl neve „vpn-toggle.sh.txt”, és csak 2,0 KiB méretű (tehát egy szempillantás alatt le kell jönnie). Azonban, semmi akadály, hogy a szkriptet alábbi leírás alapján begépeled. (Tipp: ha komolyan meg akarod tanulni a bash-szkript írását, a szkript gépelése lassabb, így jobban megértheted a parancsok működését és ezeknek a parancsoknak a felyes formátumát.)

Amint azt korábban más, a magazinban bemutatott szkripteknél mondtam, tárold a szkriptet abban a könyvtárban, ahová általában a bash szkripteket mented. Távolítsd el a „.txt” fájlkiterjesztést, és tedd végrehajthatóvá a fájlt. Remélhetőleg a könyvtár, ahová a bash szkripteket mented a számítógép \$PATH leírásában szerepel. Ha igen, akkor csak annyit kell tenni, hogy beírod a szkript nevét (vpn-toggle.sh) a szkript elindításához. Ellenkező esetben szkript teljes elérési útját meg kell adni a futtatáshoz.

Tehát, íme a vpn-toggle szkript. Kicsit hosszabb a szokásosnál, mert igyekszem jobban kiegészíteni a szkript különböző részeinek működésével kapcsolatos megjegyzésekkel. A szkript tényleges „működő” része csak 10 sornyi bash kód.

1. `#!/bin/bash`
2. `#`
3. `# VPN Toggle script.`
4. `# Written by Paul Arnote`
5. `# Read the article in the June 2026 issue of The PCLinuxOS Magazine`
6. `# https://pclosmag.com/`
7. `#`

Két új szkript az OpenVPN-kapcsolat időszakos ki- és bekapcsolására

```
8. # This script will toggle the currently running VPN off as selected by
9. # the end user from the menu attached to an icon placed in the user's
10. # notification tray. The VPN can be turned off or on, depending on
11. # which menu item the end user selects.
12. #
13. # Requirements: yad, Network Manager
14. #
15. # This script is released under the GPL v2.0 license, and may be
16. # freely distributed under the terms of the GPL v2.0 license.
17.
18. # Debugging line ... uncomment the "set -x" line below to see
19. # a "trace" of the script actions.
20. # set -x
21.
22. # Get the active VPN connection name
23. ActiveVPN=$(nmcli -g NAME,TYPE con show --active | grep vpn)
24.
25. # Extract just the VPN name (before the colon)
26. ActiveVPN2=$(echo "$ActiveVPN" | cut -d ':' -f1)
27.
28. # Check to see that the VPN is already running. If it isn't, display
29. # a warning message to the user, and exit the script. The user's VPN
30. # must be running before launching this script.
31. if [[ -z "$ActiveVPN2" ]]; then
32. yad --warning --width=350 --height=200 --text-align=center --
button=Cancel:1 --title="VPN Toggle" --text="\n\nVPN appears to not be
running.\n\nVPN <b>*must*</b> be running before\n\nlaunching this script.
\n\nLaunch/start your VPN,\n\nand run this script again.\n\nExiting script.
\n\nThis message will\n\nself destruct in 10 seconds." --timeout=10 --
timeout-indicator=bottom
33. if [[ $? == 1 || $? == 70 ]]; then
34. exit
35. fi
36. fi
37.
38. # Using YAD, we create an icon in the notification tray, and assign
39. # a menu to it with only 3 items: Turn VPN Off, Turn VPN On, and Quit.
40. # The user selects the appropriate menu choice, and either turns the
```

```
41. # VPN off, on, or quits the script.
42. yad --use-interp --notification \
43. --listen --image="networking_section" \
44. --text="VPN Toggle" \
45. --menu="Turn VPN Off!nmcli con down id $ActiveVPN2!
networking_section \
46. |Turn VPN On!nmcli con up id $ActiveVPN2!netorking_section \
47. |Quit!quit!gtk-quit"
48.
49. # Exit the script gracefully
50. exit 0
```

1. **sor** Természetesen a bash „she-bang”, ami elindítja a dolgokat. A 2-19. **sor** megjegyzések, és magyarázatot nem igényel. A 20. **sor** (a szkriptben kikommentelve) a „set -x” sor a szkript készítésekor a hibakeresés miatt került be. Itt maradt, arra az esetre, ha szeretnéd (vagy szükséged lenne rá) látni a kimenetet a terminál munkamenetében.



A 23. **sorral** lekérdezzük az aktív VPN kapcsolatot a **nmcli -g NAME,TYPE con show --active | grep vpn** paranccsal, és tárolja ActiveVPN változóként. A visszaadott aktív VPN-kapcsolat nem használható „így ahogy van,” tehát a 26. **sorban**, leválasztjuk az ActiveVPN változó „: vpn” részét, és ActiveVPN2-ként tároljuk, hogy csak az aktív VPN-kapcsolat nevét kapjuk, amit használhatunk.

A 31-36. **sor** ellenőrzi, hogy a VPN fut-e, és ha nem, akkor hibaüzenetet ad (l. fent). Ez a hibaüzenet mondja a felhasználónak, hogy indítsa el VPN-t, majd futtassa újra a szkriptet. A hibaüzenet egy yad párbeszédpanelben jelenik meg, aminek a „timeout” (lejárati) értékre 10 másodperc; a párbeszédpanel alján pedig egy

Két új szkript az OpenVPN-kapcsolat időszakos ki- és bekapcsolására

visszaszámláló jobbról balra mozogva mutatja, a futó 10 másodperc visszaszámlálás állását. Amikor az időzítő lefutott, a párbeszéd automatikusan bezárul, és a szkript kilép.

Tehát a VPN-nek ****futnia kell**** a szkript elindítása előtt. Ellenkező esetben nincs mód az aktív VPN-kapcsolat nevének „tárolására” az ActiveVPN és az ActiveVPN2 változóiban. A szkript nem közvetlenül ellenőrzi, hogy aktív-e a VPN-kapcsolat. Azt nézi, hogy a \$ActiveVPN2 egy üres karakterlánc-e, ami azt mutatná, hogy a VPN-kapcsolat nem élt a szkript indításakor, és ez elég jó.



A **42-47. sor** állítja be a menüt az értesítési területen (fent látható). Miután jobb egérgérintéssel kiválasztod az ikont az értesítési területen, válaszd a „VPN Off”-ot a VPN kikapcsolásához, és a „VPN On” opciót a bekapcsoláshoz. Természetesen a „Quit” menüből történő kiválasztása a szkript kilépését okozza. Végül, az **50. sorral** a szkript kecsesen kilép.

Mellékesen megjegyezném, a vpn-toggle akkor is működik, ha wifi-hálózatot váltasz. Az általam gyakran használt vezeték nélküli routeren két hálózat van. Az egyik a 2.4 GHz-es sávban, a másik pedig az 5,0 GHz-es sávban működik, és mindkettő úgy van beállítva, hogy ugyanazt a VPN-kiszolgáló kapcsolatot használja. Szóval, amikor vezeték nélküli kapcsolatot cseréltem, minden gond nélkül működött. De még ha nem is használnák ugyanazt a VPN-kiszolgálót, a vpn-toggle továbbra is működnie kell. Csak az előző hálózat VPN-kiszolgálójához csatlakozik

Hogy teljesen őszinte legyek, nem tártam fel teljesen ennek a felállásnak az lehetőségét, de a logika azt diktálja, hogy nem lehet gond. Nem tudom elképzelni a módját, hogy minden lehetőséget kipróbáljak, de – számomra

– úgy működik, ahogy leírtam. Ami még jobb, hogy a VPN-kapcsolatom (ahogy az esetek 98%-ában) aktív volt a vezeték nélküli hálózat váltásakor.

Ha a bash kód egy része ismerősnek tűnik, gratulálok az éles szemhez, illetve a jó memóriához. A menükészítő kódot a [játékindító](#) szkriptemből „kölcönöztem”. Ez egy csodás lehetőség a bash szkriptek írásában ... kód az egyik szkriptről kisebb módosításokkal újra felhasználható egy másikban.

Időzített VPN-váltás

Az előző (vpn-toggle) szkript egyik elvi „hiányossága”, hogy elfelejtheti visszakapcsolni a VPN-t, így a PPI-t kíváncsi szemeknek teszi ki. Tehát kitaláltam egy másik szkriptet, ami kikapcsolja a VPN-t, majd egy megadott idő után visszakapcsolja.

Akár az előző szkriptet, ezt is [letöltheted](#) a magazin szerveréről (vagy beírhatod, ha úgy tetszik). És az előző szkriphoz hasonlóan ezt is tárolj abban a könyvtárban, ahol általában a többi bash szkriptet tartod. Távolítsd el a “.txt” fájlkiterjesztést, és tedd végrehajthatóvá a szkriptet. Erre is vonatkozik a tanács, a szkriptnek a számítógép \$PATH-változójában szereplő könyvtárba való mentésére.

Tehát, alább látható a szkript, amit „vpn-timed-toggle.sh.”-nak neveztem el. Mielőtt kielemeznénk, nézd át. Az előző szkriphoz hasonlóan ez is egy kicsit hosszabbnak tűnik a szokásosnál, mivel igyekszem „javítani” a szkriptjeim kommentelésén. De ennek a szkriptnek a „üzleti” része körülbelül 30-40 sornyi bash kód (igen, egy kicsit hosszabb, mint az előző szkript), a többi megjegyzés. Ennek ellenére a teljes szkript csak 3,7 KiB, így egy szempillantás alatt le is kell töltenie.

1. `#!/bin/bash`
- 2.
3. `# Timed VPN Toggle script.`
4. `# Written by Paul Arnote`
5. `# Read the article in the June 2026 issue of The PCLinuxOS Magazine`
6. `# https://pclosmag.com/`

Két új szkript az OpenVPN-kapcsolat időszakos ki- és bekapcsolására

```
7. #
8. # This script will toggle the currently running VPN off for a specified
9. # amount of time, and then restart that same VPN when the timer has
10. # elapsed.
11. #
12. # Requirements: zenity, yad, Network Manager, notify-send
13. #
14. # This script is released under the GPL v2.0 license, and may be
15. # freely distributed under the terms of the GPL v2.0 license.
16
17. # Debugging line ... uncomment the "set -x" line below to see
18. # a "trace" of the script actions.
19. # set -x
20.
21. # Get the active VPN connection name
22. ActiveVPN=$(nmcli -g NAME,TYPE con show --active | grep vpn)
23.
24. # Extract just the VPN name (before the colon)
25. ActiveVPN2=$(echo "$ActiveVPN" | cut -d ':' -f1)
26.
27. # Check to see that the VPN is already running. If it isn't, display
28. # a warning message to the user, and exit the script. The user's VPN
29. # must be running before launching this script.
30. if [[ -z "$ActiveVPN2" ]]; then
31. yad --warning --width=350 --height=200 --text-align=center --
button=Cancel:1 --title="Timed VPN Toggle" --text="\n\nVPN appears to not
be running.\n\nVPN <b>*must*</b> be running before\n\nlaunching this
script. \n\nLaunch/start your VPN,\n\nand run this script again.\n\nExiting
script. \n\nThis message will\n\nself destruct in 10 seconds." --timeout=10 --
timeout-indicator=bottom
32. if [[ $? == 1 || $? == 70 ]]; then
33. exit
34. fi
35. fi
36.
37. # Get the number of hours to toggle VPN off, and convert to minutes
38. hrs=$(zenity --entry --title="Timed VPN Toggle" --text="Enter the
number of hours for your timer...")
39. if [ $? == 1 ]; then
40. exit
41. fi
42. if ! [[ "$hrs" =~ ^[0-9]+$ ]]; then
43. zenity --error --title="Timed VPN Toggle" --text="Hours must be a
number."
44. exit
45. fi
46. hours=$(( $hrs * 60 ))
47.
48. # Get the number of minutes for the VPN to be toggled off.
49. # The number of minutes must be between 0 and 59 minutes, or an error
50. # message will be displayed and the script will exit.
51. mins=$(zenity --entry --title="Timed VPN Toggle" --text="Enter the
number of minutes for your timer...\n\nNumber MUST be between 0 and
59.")
52. if [ $? == 1 ]; then
53. exit
54. fi
55. if ! [[ "$mins" =~ ^[0-9]+$ ]]; then
56. zenity --error --title="Timed VPN Toggle" --text="Minutes must be
a number."
57. exit
58. fi
59. if ! [[ "$mins" =~ ^[0-9]+$ ]] || (( mins < 0 || mins > 59 )); then
60. zenity --error --title="Timed VPN Toggle" --width=200 --height 100
--error --text="Exiting.\n\nNumber entry\n\nout of range.\n\nMust be a
number\n\nbetween 0 & #38; 59."
61. exit
62. fi
63.
64. # Add the hours (converted to minutes) to the minutes specified by
65. # the user for the total number of minutes of the timer.
66. TIME=$(( $hours+$mins ))
67.
68. # Format the "message" text for the notification that is displayed
69. msg=$(printf "$ActiveVPN2 OFF for the next\n%02d Hours %02d
Minutes")
```

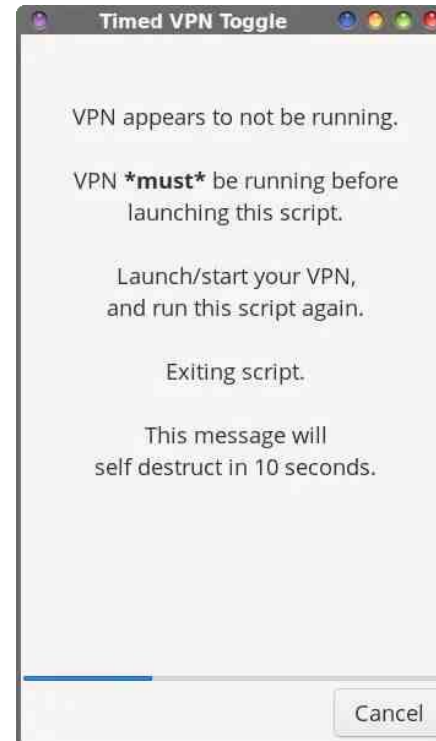
Két új szkript az OpenVPN-kapcsolat időszakos ki- és bekapcsolására

```
$hrs $mins)
70.
71. # Turn off the currently connected VPN, and display the notification
72. # that the VPN has been turned off for the specified amount of time
73. nmcli con down id "$ActiveVPN2"
74. notify-send --icon=networking_section "Timed VPN Toggle" "$msg"
75.
76. while [ $TIME -gt 0 ]
77. do
78. sleep 60
79. TIME=$(( $TIME - 1 ))
80. if [[ $(($TIME)) -gt 0 ]]; then
81. notify-send --icon=networking_section "Timed VPN Toggle"
"$ActiveVPN2 will reconnect in $TIME minutes"
82. fi
83. done
84.
85. # Since the timer elapsed, turn the previously connected VPN back on
86. # and display a notification to the user that the VPN has been turned
87. # back on
88. nmcli con up "$ActiveVPN2"
89. msg=$(printf "Timed VPN Toggle timer has elapsed.\n$ActiveVPN2 VPN
connection restored.")
90. notify-send --icon=networking_section "Timed VPN Toggle" "$msg"
91.
92. # Exit the script gracefully
93. exit 0
```

A tipikus bash “she-bang” indítja a dolgokat az **1. sorban**. A megjegyzések **2-18. sorba** kerültek, és magától értetődőnek kell lenniük. A **19. sorban** (ami ki van kommentelve a szkript fenti leírásában), a „set -x” parancs értékes visszajelzést adhat a szkript hibakereséséhez. Ha megnéznéd a működés során a szkript kimenetét, egyszerűen töröld a sor elejéről a „#”-t.

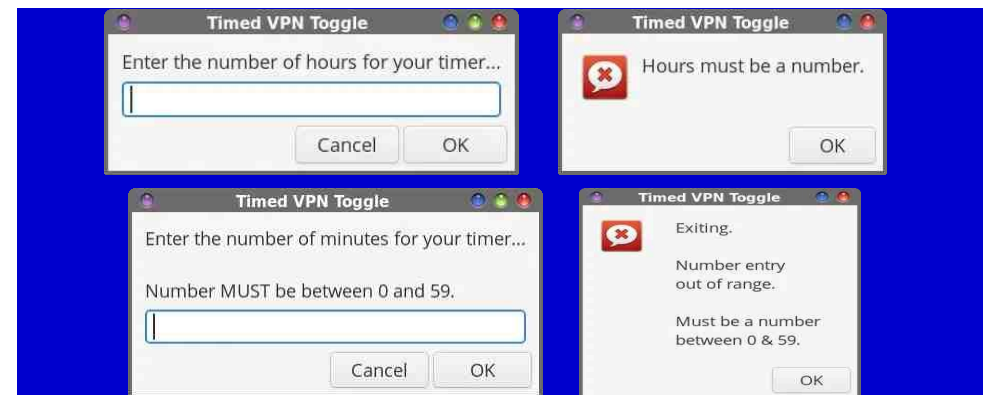
A **22. sor** meghívja az **nmcli -g NAME,TYPE con show --active | grep vpn** parancsot az ActiveVPN változó feltöltésére az aktív VPN-kapcsolat nevével. Azonban nem használhatjuk úgy, ahogy van ([vpn-kapcsolat neve]: vpn), ezért az aktív VPN-kapcsolat nevének „vpn” részét

leválasztjuk, és csak az első részt (a tényleges nevet) rendeljük hozzá a ActiveVPN2 változóhoz **25. sor**.



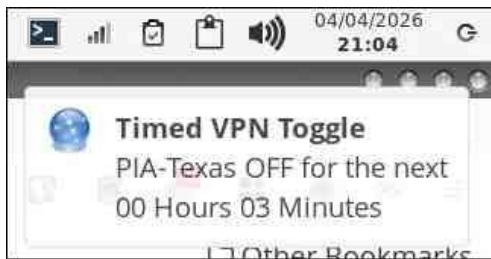
A **30-35. sor** ellenőrzi, hogy a VPN fut-e, és ha nem, akkor hibaüzenetet ad (fent). A hibaüzenet megmondja a felhasználónak, hogy indítsa el VPN-jét, majd próbálja újra a szkriptet. A hibaüzenet egy yad párbeszédpanelben jelenik meg, aminek a „timeout” (lejárat) értékre 10 másodperc, a párbeszéd alján pedig egy visszaszámláló jobbról balra mozogva mutatja a 10 másodperc visszaszámlálás állását. Amikor az időzítő lefutott, a párbeszéd automatikusan bezárul, és a szkript kilép.

Tehát a VPN-nek ****futnia kell**** a szkript elindítása előtt. Ellenkező esetben nincs mód az aktív VPN-kapcsolat nevének „tárolására” az ActiveVPN és az ActiveVPN2 változóban. A szkript nem közvetlenül ellenőrzi, hogy aktív-e a VPN-kapcsolat. Azt nézi, hogy a \$ActiveVPN2 üres karakterlánc-e, ami azt mutatná, hogy a VPN-kapcsolat nem élt a szkript indításakor, és ez elég jó.



Két új szkript az OpenVPN-kapcsolat időszakos ki- és bekapcsolására

A **38-41. sor** Zenity párbeszédpanel jelenít meg, a VPN szüneteltetése órái számának megadásához. A **43-45. sorban** gondoskodunk arról, hogy a bevitt érték valóban szám legyen, és ha nem, hibaüzenettel kilépünk a szkript-ből. A **46. sorban** az órák értékét percekre konvertáljuk. A **51-62. sorban** egy másik Zenity párbeszédpanelünk van, ahol a felhasználó beírhatja a VPN szüneteltetés percei számát. A szám 0 és 59 között ****kell**** legyen különben hibaüzenet jelenik meg (egy másik Zenity párbeszédpanelen keresztül), és a szkript kilép. Az **66. sor-ban** hozzáadjuk a „minutes” (percek) párbeszédben megadott percek számát a pecre konvertált órákhoz, hogy megkapjuk a VPN szüneteltetésének teljes idejét (percben).

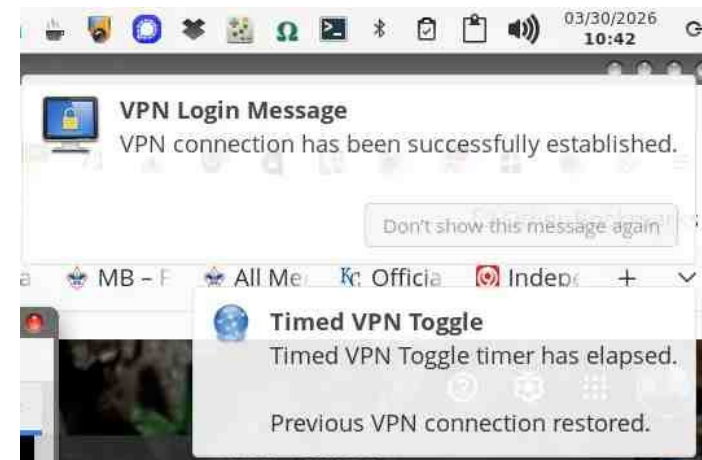


A megjelenő üzenet szövegét meg kell formálnunk, amiről a **69. sor** gondoskodik nekünk. A **73. sor** az aktuálisan futó VPN-t leállítja, miközben a **74. sor** megjeleníti az üzenetet. A **76-83. sorok** ahol egy nagyon hatékonyidőzítőt hozunk létre. Ebben a „while ... do” ciklus egészen addig fut,amíg a \$TIME változó értékenagyobb nullánál. A „while ... do” minden egyes ciklusánál 60 másodpercet „alszik” és a \$TIME változó értéke egyel csökken.



A while ... do-ba beágyazva van egy if ... then feltételes utasítás. Amíg a \$TIME értéke nagyobb nullánál, percenként megjelenik egy értesítés, ami emlékeztet a szkript futására. Az eredeti verzióban ez még nem volt, ami nagyon zavart, hogy nem tudtam emlékeztetni a felhasználót, hogy a VPN ki van kapcsolva. Persze, kereshetné a pirinyó lakatot a wifi jelerősség jelzőjén, de olyan kicsi, túl könnyű nem észrevenni. Megpróbáltam egy

ikont elhelyezni az értesítési tálcán, de (egyelőre) nem nem működik úgy, ahogy szeretném. Ráadásul van ***valami*** yad notification parancs kiadásában, ami megakadályozza a szkript lefutását (azaz a VPN újraindítását). A yad notification parancsról kevés információ van, és legjobb esetben is nehezen elérhető, ezért a „B tervre” váltottam, ami percenként megjelenít egy értesítést, ahogy az időzítő visszaszámol. Az utolsó, amit szeretnék, hogy valaki elfelejtse, hogy a VPN-je jelenleg ki van kapcsolva, és ezzel veszélyezteti a PPI-jét (attól függően, hogy mit csinál). Számomra ez tűnt a legjobb kompromisszumnak. Véleményem szerint az értesítés minimálisan zavaró, de elég észrevehető ahhoz, hogy emlékeztesse a felhasználót arra, hogy a VPN ki van kapcsolva.



Amint az időzítő eléri a nullát, a **88. sor** újraindítja a korábban aktív VPN kapcsolatot, és a **89. sor** megformálja a **90. sorban** felugró értesítést arról, hogy az előző VPN-kapcsolat helyreállt. Végül, a **93. sor** lehetővé teszi, hogy a szkript tisztán kilépjen, amikor elkészült.

A vpn-toggle-lel ellentétben nem fog rendesen működni, ha a szkript futása közben váltasz vezeték nélküli hálózatot. Az Wi-Fi hálózataim (a NetworkManageren keresztül) úgy vannak beállítva, hogy automatikusan csatlakozzanak egy VPN-hez aktiváláskor. Tehát, ha a vpn-timed-toggle futása közben módosítod, a vezeték nélküli hálózat a szokásos módon csatlakozik, és a VPN-kiszolgálót a vpn-timed-toggle szkripttől függetlenül újraindítja. A vpn-timed-toggle szkript nem figyel, hogy történt-e bármilyen változás, és azt várja, hogy a VPN-kapcsolat ki legyen

Két új szkript az OpenVPN-kapcsolat időszakos ki- és bekapcsolására

kapcsolva az indítás után (és hogy a VPN-kapcsolatod be legyen kapcsolva a szkript indítása előtt). Így valószínűleg problémák merülnek fel, mivel a vpn-timed-toggle azt várja, hogy a VPN-kapcsolat nem működik, pedig már működik.

Ha azonban több vezeték nélküli hálózatod van, és mindegyik ugyanazt a VPN-kiszolgálót használja, akkor valójában megúszhatod. Túl sok változó van ahhoz, hogy ezt teljes mértékben teszteljem, ezért szeretném már az elején felhívni a figyelmedet erre a lehetséges buktatóra. A megfelelő működés biztosítása érdekében a legjobb, ha nem váltasz vezeték nélküli hálózatot, amíg a vpn-timed-toggle szkript fut.

Ha észrevetted, újra felhasználtam néhány bash kódot ebben a szkriptben. A szkript időzítő része „kölcsonzött” a [korábbi](#), az xscreensaver funkciót meghatározott ideig kikapcsoló szkriptemből. A szkript további részeit a cikk elején található vpn-toggle szkriptből „kölcsonöztem”. A bizonyítottan működő és megbízható bash kód újrafelhasználása segít a bash szkript „gyorsított” létrehozásban, és ebben az esetben biztosan jól működött.

Ha csak néhány percre szeretnéd vagy kell szüneteltetni a VPN-kapcsolatot, nyugodtan írd be „0”-t az órákhoz, majd a percek számát a „minutes” párbeszédpanelen. Ennek a néhány percnak a végén a VPN-kapcsolat helyreáll.

A vpn-timed-toggle szkriptben az a szép, hogy nem kell aggódni a VPN-kapcsolat újraindítása miatt, mivel a szkript automatikusan visszaállítja neked.

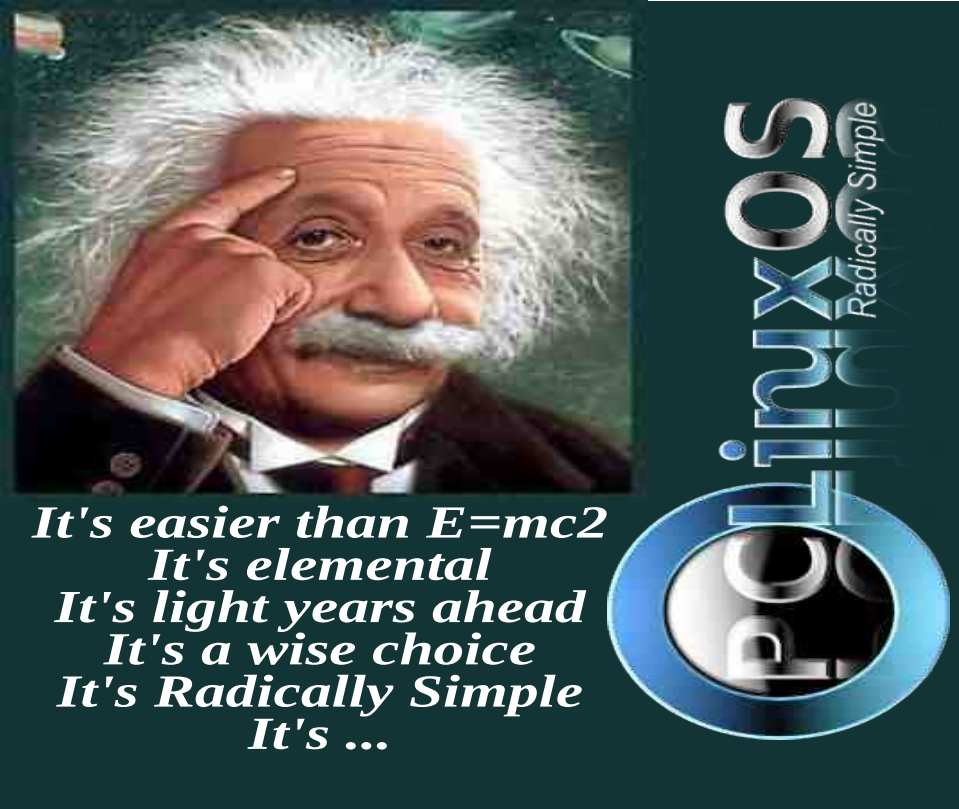
Összegzés

Az itt bemutatott két szkript közül a rövidebb „vpn-toggle”-t volt a nehezebb megírni. Először is ott olyan utat tapostam ki, amin még soha nem jártam. Másodszer, a yad dokumentációja nem a legjobb, ezért eltartott egy ideig, amíg kitaláltam a menü megfelelő formázását, hogy a parancsokat megfelelően hajtsa végre. És őszintén szólva, sokkal jobban

ismerem a Zenityt, mint a yadot. Miután elkészült a „vpn-toggle” szkript, a „vpn-timed-toggle” szkript létrehozása csak néhány percet vett igénybe. Sőt, tökéletesen futott, amikor először kipróbáltam.

Március vége óta futtatom ezeket a szkripteket (főleg a „vpn-toggle” verziót), és még nem mondtak csődöt. Kötelességtudóan szüneteltetik a VPN-kapcsolatomat, amikor kérem, és visszaállítják a VPN-kapcsolatomat, amikor kérem. Végző soron erre törekedtem. Most a VPN-em szüneteltetése, és visszaállítása csupán pár egyszerű egérgattintás kérdése nélkül, hogy a NetworkManager rendetlen és kényelmetlen menüjében kellene navigálni.

Szerkesztő megjegyzése: maradj velünk! Jövő hónapban bemutatjuk Cuig szkriptjét, amellyel kezelheted a Wireguard VPN-kapcsolatot.



PCLinuxOS
Radically Simple

*It's easier than $E=mc^2$
It's elemental
It's light years ahead
It's a wise choice
It's Radically Simple
It's ...*