

Bízd Sherlock-ra, mutassa meg, a felhasználóneved hol van használatban

PCLinuxOS Magazine - 2024. október

Írta: Paul Arnote (parnote)

Minden hónapban szó szerint feltúrom az Internetet, témákat keresve a Magazine számára. Amikor a Liferhacker-en rátaláltam erre a [cikkre](#), megfogott és elhatároztam, hogy megnézem, át lehet-e venni a szkriptet a PCLinuxOS-felhasználók számára.



A Sherlock Project egy parancssorból (terminál alól) futó Python-szkript, ami (az írás pillanatában, a Facebook és az X kivételével) 407 „közösségi oldalt” néz végig, megadott felhasználónevet keresve. Jól jöhet a segítsége, ha újra meg kell találnod azokat a regisztrációidat, amikről talán elfeledkeztél. Esetleg abban is segíthet, hogy fiók létrehozása előtt ellenőrizd, hogy az adott oldalon, vajon a kívánt felhasználónév használatban van-e már, vagy sem. Vagy segíthet, hogy kielégítsd vele a kíváncsiságodat.

A cikk (szkriptre koncentráló témája mellett) egyik sora megragadta tekintetemet. Íme:

„Linux-felhasználóknak a dolgok egyszerűbbek: lehet, hogy a Sherlock már megtalálható a csomagkezelőben.”

Így hát, belenéztem a PCLinuxOS-tárolójába és kiderült a Sherlock NINCS a PCLinuxOS-csomagok között. Ez volt az első probléma, amivel szembekerültem.

Mielőtt továbblépnénk, ide kívánczok egy rövid megjegyzés. Miközben a PCLinuxOS-ben „alapszabály”, hogy CSAK a PCLinuxOS hivatalos tárolóiból telepítünk programokat, ennek a Python-szkriptnek a telepítése NEM károsítja a PCLinuxOS-rendszeredet. Vagyis, a saját felelősségedre folytatod (de biztos lehetsz benne, hogy nincs semmi látható veszélye).

Szerencsére a Sherlock-projekt a legfrissebb verziót a GitHub-on tartja. Elmentem hát a GitHub-hoz, hogy [letöltsem](#) a projekt forráskódját. Ha már letöltötted a forráskódot, a ZIP-fájlt csomagold ki a saját könyvtárába. Én a Zip-fájlt a „Letöltések” könyvtárba mentettem le, majd a forráskódot ott, egy alkönyvtárba bontottam ki. Terminálban beléptem abba az alkönyvtárba. Amikor ott vagy, fogd vissza magad, és ne ásd bele magad az alkönyvtár könyvtárszerkezetébe. Az alkönyvtár felső szintjén kell maradnod. Én megtartottam a könyvtár **sherlock-master** nevét, ami megegyezik a GitHub-ról letöltött ZIP-fájl nevével.

Ezután ellátogattam a Sherlock Project [honlapjára](#). A telepítési (Installation) résznél a pipx használatát javasolja a python-szkript számítógépre telepítéséhez. Vissza a PCLinuxOS-tárolójához, megnéztem van-e pipx. Ismét csak NEM volt a PCLinuxOS-tárolójában (legalábbis én nem találtam meg).

Hogy lássam, esetleg telepítve van-e, begépeltem a parancssorba, hogy „pip” és lenyomtam a Tab-billentyűt. Az összes „pip”-pel kezdődő program megjelent. A



Bízd Sherlock-ra, mutassa meg, a felhasználóneved hol van használatban

telepítő oldal azt állította, hogy a Sherlock telepítésére a pip is használható. A gépem ott volt, de az sem működött.

terminálban azt is észrevettem, hogy a **pip3** már telepítve van a gépemre. Semmit sem veszíthetek alapon, újrapróbáltam a telepítést pip3 parancs használatával. Íme a parancs:

```
pip3 install --user sherlock-project
```

Meglepetésemre működött. Itt látható a terminál kimenete, amit a Sherlock telepítése mutatott a pip3 használata során. A --user kapcsoló utasítja a Python-t, hogy a jelenlegi felhasználónak telepítse.

```
[parnote@localhost sherlock-master]$ pip3 install --user sherlock-project
Collecting sherlock-project
  Downloading sherlock_project-0.15.0-py3-none-any.whl (35 kB)
Requirement already satisfied: certifi>=2019.6.16 in /usr/lib/python3.10/site-packages (from sherlock-project) (2021.10.8)
Collecting Coloma<0.5.0,>=0.4.1
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Requirement already satisfied: requests<3.0.0,>=2.22.0 in /usr/lib/python3.10/site-packages (from sherlock-project) (2.27.1)
Collecting openpyxl<4.0.0,>=3.0.10
  Downloading openpyxl-3.1.5-py2.py3-none-any.whl (250 kB)
 250.9/250.9 kB 1.6 MB/s eta 0:00:00
Collecting requests-futures<2.0.0,>=1.0.0
  Downloading requests_futures-1.0.1-py2.py3-none-any.whl (7.6 kB)
Collecting stem<2.0.0,>=1.8.0
  Downloading stem-1.8.2.tar.gz (2.9 MB)
 2.9/2.9 MB 3.2 MB/s eta 0:00:00
Preparing metadata (setup.py) ... done
Collecting pandas<3.0.0,>=1.0.0
  Downloading pandas-2.2.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.0 MB)
 13.0/13.0 MB 10.8 MB/s eta 0:00:00
Requirement already satisfied: PySocks<2.0.0,>=1.7.0 in /usr/lib/python3.10/site-packages (from sherlock-project) (1.7.1)
```

```
Collecting et-xmlfile
  Downloading et_xmlfile-1.1.0-py3-none-any.whl (4.7 kB)
Collecting python-dateutil>=2.8.2
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
 229.9/229.9 kB 7.4 MB/s eta 0:00:00
Collecting numpy>=1.22.4
  Downloading numpy-2.1.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.3 MB)
 16.3/16.3 MB 17.3 MB/s eta 0:00:00
Collecting tzdata>=2022.7
  Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
 345.4/345.4 kB 10.9 MB/s eta 0:00:00
Collecting pytz>=2020.1
  Downloading pytz-2024.2-py2.py3-none-any.whl (508 kB)
 508.0/508.0 kB 9.3 MB/s eta 0:00:00
Requirement already satisfied: urllib3<1.27 in /usr/lib/python3.10/site-packages (from requests<3.0.0,>=2.22.0->sherlock-project) (1.26.9)
Requirement already satisfied: charset_normalizer==2.0.0 in /usr/lib/python3.10/site-packages (from requests<3.0.0,>=2.22.0->sherlock-project) (2.0.12)
Requirement already satisfied: idna>=3 in /usr/lib/python3.10/site-packages (from requests<3.0.0,>=2.22.0->sherlock-project) (3.3)
Requirement already satisfied: six>=1.5 in /usr/lib/python3.10/site-packages (from python-dateutil>=2.8.2->pandas<3.0.0,>=1.0.0->sherlock-project) (1.16.0)
Using legacy 'setup.py install' for stem, since package 'wheel' is not installed.
Installing collected packages: stem, pytz, tzdata, python-dateutil, numpy, et-xmlfile, colorama, requests-futures, pandas, openpyxl, sherlock-project
  Running setup.py install for stem ... done

WARNING: The scripts f2py and numpy-config are installed in '/home/parnote/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
```

Bízd Sherlock-ra, mutassa meg, a felhasználóneved hol van használatban

```
WARNING: The script sherlock is installed in '/home/parnote/.local/bin' which is not on PATH.
```

```
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
```

```
Successfully installed colorama-0.4.6 et-xmlfile-1.1.0 numpy-2.1.1 openpyxl-3.1.5 pandas-2.2.2 python-dateutil-2.9.0.post0 pytz-2024.2 requests-futures-1.0.1 sherlock-project-0.15.0 stem-1.8.2 tzdata-2024.1
```

A program alapból a ~/.local/bin könyvtárba települ. Mivel az a könyvtár NINCS az útvonalleírásomban (path), bármikor ha futtatni akarom a Sherlock-ot, be kellett volna lépnem abba a könyvtárba. Ahogy a következő kimenet is mutatja, nagyon jól működik.

Ha a Sherlock olyan könyvtárba települ, ami NINCS az útvonalleírásodban, oda kell belépned és be kell írnod, hogy ./sherlock [felhasználónév] a parancssorba. A [felhasználónév] természetesen lecserélhető a keresett elnevezésre. A lenti példához a ./sherlock parnote -ot írtam be, hogy rákeressek a szokásos felhasználónevemre.

Hogy „megszabaduljak” attól a kényelmetlenségétől, hogy minden alkalommal, ha futtatni akarom a parancsot, be kell lépni a ~/.local/bin könyvtárba, egyszerűen átvittem a sherlock-szkriptet a ~/bin könyvtárba, ami az útvonalam része. Most már a sherlock [felhasználónév] a parancssorba begépelése elindítja a szkriptet anélkül, hogy a telepítési könyvtárba váltással kellene izmoznom, és az indító elé be kellene gépelnem a ./-t. Említésre érdemes, hogy készíthető szimbolikus hivatkozás a sherlock szkriptfájltra egy olyan könyvtárban, ami része az útvonalleírásodnak. Én nem láttam értelemét, mivel a sherlock-szkript (legalábbis a programindító része) nagyon kicsi, mindössze 224 byte méretű.

```
[parnote@localhost bin]$ ./sherlock parnote
```

```
[ ] Checking username parnote on:
```

```
[+] 8tracks: https://8tracks.com/parnote
[+] AllMyLinks: https://allmylinks.com/parnote
[+] Amino: https://aminoapps.com/u/parnote
[+] Codeberg: https://codeberg.org/parnote
```

```
[+] DailyMotion: https://www.dailymotion.com/parnote
[+] Dealabs: https://www.dealabs.com/profile/parnote
[+] Disqus: https://disqus.com/parnote
[+] Fiverr: https://www.fiverr.com/parnote
[+] Freesound: https://freesound.org/people/parnote/
[+] GitHub: https://www.github.com/parnote
[+] Gitee: https://gitee.com/parnote
[+] HackTheBox: https://forum.hackthebox.eu/profile/parnote
[+] HackenProof (Hackers): https://hackenproof.com/hackers/parnote
[+] HackerEarth: https://hackerearth.com/@parnote
[+] HudsonRock: https://cavalier.hudsonrock.com/api/json/v2/osint-tools/search-by-username?username=parnote
[+] Instructables: https://www.instructables.com/member/parnote
[+] Issuu: https://issuu.com/parnote
[+] Kick: https://kick.com/parnote
[+] Launchpad: https://launchpad.net/~parnote
[+] LibraryThing: https://www.librarything.com/profile/parnote
[+] Lichess: https://lichess.org/@/parnote
[+] NitroType: https://www.nitrotype.com/racer/parnote
[+] Pastebin: https://pastebin.com/u/parnote
[+] ProductHunt: https://www.producthunt.com/@parnote
[+] Roblox: https://www.roblox.com/user.aspx?username=parnote
[+] Scribd: https://www.scribd.com/parnote
[+] Snapchat: https://www.snapchat.com/add/parnote
[+] Spotify: https://open.spotify.com/user/parnote
[+] Strava: https://www.strava.com/athletes/parnote
[+] TLDR Legal: https://tldrlegal.com/users/parnote/
[+] Telegram: https://t.me/parnote
[+] Venmo: https://account.venmo.com/u/parnote
[+] Vero: https://vero.co/parnote
[+] Wikipedia: https://en.wikipedia.org/wiki/Special:CentralAuth/parnote?uselang=qqx
[+] WordPress: https://parnote.wordpress.com/
[+] Xbox Gamertag: https://xboxgamertag.com/search/parnote
[+] YouTube: https://www.youtube.com/@parnote
[+] last.fm: https://last.fm/user/parnote
```

```
[*] Search completed with 38 results
```


Bízd Sherlock-ra, mutassa meg, a felhasználóneved hol van használatban

A Sherlock csak a találatokat listázza ki. A fenti találatok között vannak, amikről fogalmam SINCS, egyáltalán milyen oldalak, és nem tudok visszaemlékezni, hogy valaha megnéztem volna. Soha. A többit ismerem és néhányon tényleg van fiókom. Van néhány, amiről tudom, hogy kihez tartozik, még ha nem is az enyém. Például, egyáltalán NINCS Venmo-fiókom, Ugyanakkor, kiír egy parnote felhasználónevet. A hivatkozást megnyitva látszik, hogy az a nagynéném, és ott a felhasználóneve parnote. Snapchat-fiókom sincs. Valakihez tartozik, akit nem ismerek.

Többféleképpen megnézheted a Sherlock által jelzett oldalakat. Talán a legnehezebb, ha egyenként begépeljük a kapott címeket a böngészőbe. Beletelik egy kis időbe, a kapott eredmények száma függvényben. Legegyszerűbb, az egyes oldalak nevére jobb billentyűvel kattintani és a menüből megnyitni. Az alapbeállítás szerinti böngészőben nyílik meg. A harmadik lehetőség, hogy a terminál ablakában a kapott eredményt kijelölöd és a felhasználói menüből a „Másolás HTML-ként” választva kimásolod. Illeszd be egy szövegszerkesztőbe (mint a Kate, Mousepad, Geany, de NE használd a LibreOffice Writer-t) és mentsd HTML fájlként. Ezután, a HTML-fájlt megnyithatod a kedvenc webböngésződben (Fájl → Fájl megnyitása) és az összes hivatkozás ott lesz megnyitva a böngésződben anélkül, hogy ugrálgatnod kellene a böngésző és a terminál ablaka között.

Több felhasználónévre kellene keresned? Szerencsére a Sherlock lehetővé teszi több név használatát parancssori argumentumként. Például, szeretnék a

parnote és a másikfelhasználó névre keresni. Beírom **sherlock parnote másikfelhasználó** a parancssorba. Noha számos felhasználónevet beírhatasz a parancssorba argumentumként, én jobb szeretem a felhasználónevek egyenkénti ellenőrzését.

Szintén, noha a Sherlock a felhasználónévre mutathat találatokat, nem jelenti, hogy azok a tied. Csak azt, hogy azokon az oldalakon megtalálta a felhasználót. A Sherlock azt sem tudja, hogy ki vagy.

Nincs megkötés a beírni szándékozott felhasználónévre vonatkozóan. Bármilyen név használható a

keresés során. Például beírtam „sherlock Texstar”-t a parancssorba és 66 pozitív megerősítés érkezett. A „sherlock meemaw” beírás 134 találatot eredményezett.

A Sherlock nagyon jól és nagyon gyorsan dolgozik. A legtöbb esetben a Sherlock kevesebb mint két perc alatt hoz eredményt és többnyire egy perc sem kell. Noha ez nem olyan eszköz, amit gyakran használsz, egyedülállóan hasznos és segíthet megtalálni a fiókokat, amikről elfelejtetted, hogy vannak (nekem segített megtalálni néhány elfelejtett fiókot).

