

Késleltetett képernyőkép készítő szkript

PCLinuxOS Magazine – 2025. december

Írta: Paul Arnote (parnote)

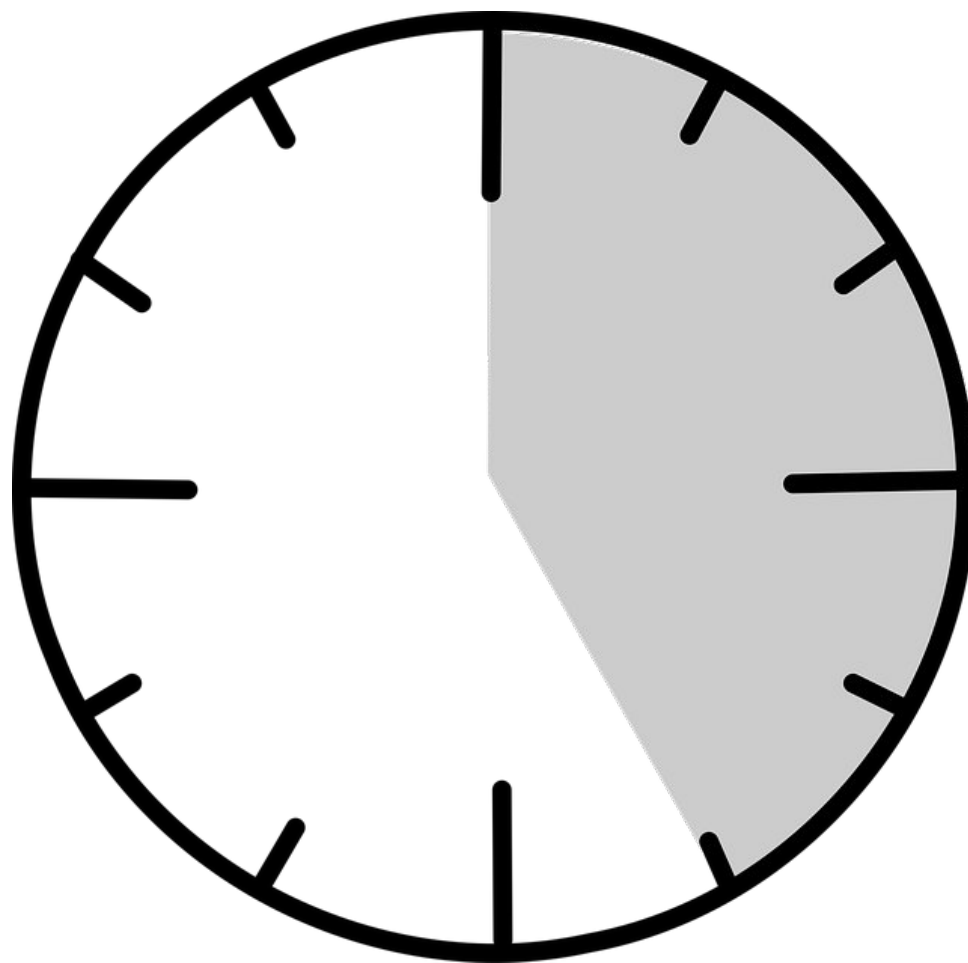
Az igazat megvallva, már egy ideje keresem a módot késleltetett képernyőképek készítésére. Talán szükség lenne némi háttérre, hogy értsd, miért van ez egy ideje a „kívánságlistámon”.

Ahogy korábban is írtam, 2022 végén mentem nyugdíjba, miután 35 évet dolgoztam regisztrált légzésterapeutaként. NEM tervezem visszatérni a munka világába. A feleségem is regisztrált légzésterapeuta. Az ágy melletti frontvonalról mesélt „háborús történeteit” hallva csak alátámasztja a döntésemet, hogy akkor visszavonultam, amikor visszavonultam. Ennek ellenére továbbra is életben tartom az engedélyemet... biztos, ami biztos. Sokkal könnyebb fenntartani, mint esetleg visszaszerezni, ha már lejárt.

A légzésterápiára vonatkozó engedélyem megőrzésének „folyamatának” része (a légzőszervi terapeuták, ápolók, orvosok és más egészségügyi szakemberek esetén hasonló az eljárás), hogy két évente teljesíteni 24 CEU-t (fenntartó továbbképzési egységet... minden egyes „egység” körülbelül egy óra oktatást jelent). Ezek felénél az engedélyezési bizottság „élő kapcsolatot” **vár el**. Ez azt jelenti, hogy vagy legalább 12 CEU-t kell elvégezni olyan környezetben, ahol kérdéseket tehet fel, és "élő kapcsolatot" tarthat az előadóval.

Kétféleképpen lehet megszerezni ezeket az „élő kapcsolattartó” CEU-kat. Szemináriumon, vagy élő webináriumokon történő online „részvétellel”. Az előbbi rendszerint megfizetendő költséggel jár (és a költségek gyorsan megemelkedhetnek), míg webináriumokat ingyenesen kínálják, ezeket a berendezések beszállítói és mások szponzorálják, akiknek érdeke a légúti oktatási információk terjesztésében. Természetesen mind a 24 CEU lehet „live contact” típusú, és ez az az út, ami mellett már évek óta elköteleztem magam.

A webináriumok többsége a Zoom-on keresztül megy. Másokat egyéni HTML5 formában kínálnak. Mindkét esetben a webináriumok végén



Kép: [OpenClipart-Vectors](#) a [Pixabay](#)-ről

megjelenik egy speciális kód, vagy egy QR-kód, amit be kell olvasni. Részt vehet a webináriumon, de ha távol van a számítógéptől, amikor a kód megjelenik, nem kaphat igazolást a részvételtől. Esetleg kiugrottál a mosdóba, vagy elmentél inni vagy enni. A visszafelé esetleg meg kell nézni, mit csinálnak a gyerekek (vagy nem csinálnak). Ez a kód lehetővé

teszi, hogy kitöltse a szükséges kérdőívet és visszajelzési űrlapot, amely jóváírja a kreditet a részvételről.

Ez a kód mindig az óra végén jelenik meg (az egyes webinariumok valójában csak 50-55 percig tartanak, de egy órát számolnak). Többször előfordult már, hogy részt vettem webinariumon, de nem kaphattam igazolást a részvételéről, egyszerűen azért, mert hiányzott a kód az előadás végén.

De arra gondoltam, talán „ütemezhetek” egy képernyőképet az óra végén vagy akörül, és akkor nem hagynám ki a kód megérkezését. Eddig minden korábbi próbálkozásom teljes kudarc volt.

Mivel Xfce asztali gépet használok, először az xfce4-screenshooter programmal próbálkoztam. De nem voltam túl sikeres a kódról képernyőkép készítésében. Először is, az xfce4-screenshooter készít egy képet, majd megáll, hogy megkérdezze, mit csináljon az általa készített képernyőkép (mentse, vágólapra másolja, felhőszolgáltatásba feltöltse stb.). Lényegében felhasználói beavatkozásra van szükség a képernyőkép mentéséhez. Ezért a képernyőkép időzítése a prezentáció végén nagyon „kérdéses”, mivel csak EGY esélyed van a kód rögzítésére. Igen. Egy. Ellenkező esetben az egész webinariumon töltött óra hiábavaló. Tehát szükségem egy módszerre, amivel a prezentáció végén, meghatározott időn keresztül képernyőképet készíthetek szabályos időközönként, hogy a kódot biztosan rögzíthessem.

Kellett valami, amivel automatikusan lehet (szkripttel vezérelve) egy sor képernyőképet készíteni a bemutató végén, és elmenti a számítógépem merevlemezére. Tehát az én „különleges esetemet” figyelembe véve ez azt jelentette, hogy meg kellett írnom a saját szkriptemet. Semmi, amit a megoldás keresése során találtam, még csak közel sem volt ahhoz, hogy képes legyen arra, amire szükségem volt.

Szerencsére korábban már írtam egy egyszerű visszaszámláló szkriptet, ami a PCLinuxOS Magazine 2024. márciusi számában jelent meg. A szkript adta meg a szükséges keretet egy saját bash szkript létrehozásához, hogy legyenek késleltetett képernyőképeim. Csak ki kellett cserélnem a szkriptnek azt a részét, ami a visszaszámláló lejártakor a riasztást adta, egy olyan kódra, ami egy sorozat képernyőképet készített.

Szóval, vessünk egy pillantást az általam kitalált szkriptre. Aztán végignézzük, hogy a szkriptben mi történik. A szkript futtatásának egyetlen követelménye, hogy a **zenity** és a **scrot** telepítve legyen a PCLinuxOS tárolóból.

```
1. #!/bin/bash
2.
3. # Delayed Screenshot script.
4. # Written by Paul Arnote
5. # Read the article in the Dec 2025 issue of The PCLinuxOS
   Magazine
6. # https://pclosmag.com/
7.
8. hrs=$(zenity --entry --title="Delayed Screenshot" --
   text="Enter the number of hours for your timer...")
9. if [ $? == 1 ]; then
10. exit
11. fi
12. hours=$(( $hrs * 60 ))
13.
14. mins=$(zenity --entry --title="Delayed Screenshot" --
   text="Enter the number of minutes for your timer...
   \n\nNumber MUST be between 0 and 59.")
15. if [ $? == 1 ]; then
16. exit
17. fi
18. if [[ ( $(($mins)) -lt 0 ) || ( $(($mins)) -gt 59 ) ]];
   then
19. zenity --error --title="Delayed Screenshot" --width=200
   --height 100 --error --text="Exiting.\n\nNumber entry\nout
   of range.\n\nMust be a number\nbetween 0 & 59."
20. exit
21. fi
22.
23. if [ ! -d $HOME/Pictures/ScreenCaps ]; then
24. mkdir $HOME/Pictures/ScreenCaps
25. fi
26. cd $HOME/Pictures/ScreenCaps/
```

```

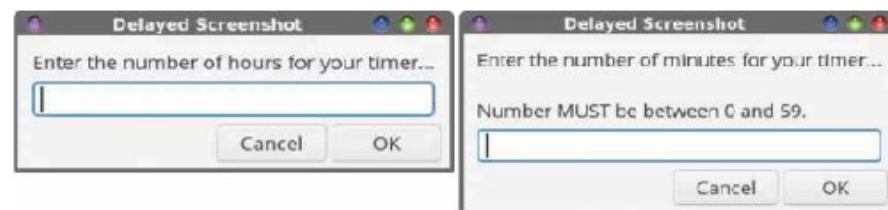
27.
28. TIME=$(( $hours+$mins ))
29.
30. while [ $TIME -gt 0 ]
31. do
32. sleep 60s
33. xscreensaver-command -deactivate >/dev/null
34. TIME=$(( $TIME - 1 ))
35. done
36.
37. for (( i=0; i<10; i++ ))
38. do
39. sleep 30s
40. scrot -z --quality 10 '%Y-%m-%d-%H-%M-%S.jpg'
41. done
42.
43. sleep 10s
44. zenity --info --title="Delayed Screenshot" --text="10
images captured, 30s apart, and\ntstored in ~/Pictures/
ScreenCaps/."
45. exit 0

```

A szkript első fele sorról sorra a visszaszámláló szkriptemből származik. Zenity párbeszédpaneleket használ, hogy a felhasználó adja meg az késleltetés idejét. A felhasználó megadja az órák számát (\$hrs), ezt percekre (\$hours) konvertálja. Ezután a felhasználó beírja a percek számát. Ha a bemenet egy szám, ami NEM a 0-59 közé esik, a szkript hibaüzenetet jelenít meg, és kilép. Ha a percek száma ezen a tartományon belül van, akkor a perceket hozzáadja a \$hours-hoz (34. sor), hogy megkapjuk a teljes késleltetési időt percekben (\$TIME, a 34. sor). Ennek megfelelően,



ha képernyőkép készítése előtt 90 perces késleltetés kell, vagy akar, nem írható be 90 perc. Az órákhoz „1”-et kell beírni, ami 60 percre lesz átszámítva, majd ehhez 30 percet kell hozzáadni, hogy 90 perc legyen.



Továbbá, kell egy hely a képernyőképek tárolására. A 23. sorban a szkript ellenőrzi, hogy létezik-e a ~/Pictures/ScreenCaps/ könyvtár. Ha nem, akkor a 24. sor létrehozza. Ezután a 26. sorban kiadjuk a „cd” parancsot, hogy a „munkakönyvtárunkra” ~/Pictures/ScreenCaps-re váltson.

Ki kell kapcsolnunk vagy felfüggeszteni a képernyővédőt is. Erre az xscreensaver „üres képernyő” funkcióját használom. Ha nem kapcsolja ki vagy deaktiválja a képernyővédőt, akkor a szkript kötelességszerűen elkészíti a képernyőképeit üres, fekete képernyőkről. A 30. sortól kezdve beállítunk egy while ... do ciklust, ami addig fut, amíg a \$TIME nagyobb, mint nulla. Ezután 60 másodpercig vár, majd kiadja a parancsot a képernyővédő kikapcsolására, és eggyel csökkenti az összes perc számát.

Ha lejárt az idő, egy for ... do ciklust indítunk a képernyőképek elkészítéséhez. A for ciklust úgy állítottam be, hogy 10 képernyőképet készítsen. Előbb várunk 30 másodpercet, majd kiadjuk a parancsot a képernyőkép elkészítésére. A scrot-ot használom a képkészítésére, ez az egyetlen segédprogram, ami automatikusan menti a képernyőképet felhasználói beavatkozás nélkül. A „-z” paraméter azt mondja a scrot-nak, hogy csendben készítse el ezeket a képernyőképeket. Enélkül NAGYON bosszantó hangot ad ki, jelezve, hogy képernyőképet készít.

A "--quality 10" paraméterrel a képek minőségét nagyon alacsonyra állítom. Nem teszem közzé ezeket a képeket. Csak el kell tudnom olvasni. A „--quality 10” paraméter nélkül a 10 kép egyenként több mint 6 MiB méretű lenne. A „--quality 10” paraméter használatával (a scrot kényszerítésével a képek JPG-fájlként való mentéséhez) mind a 10 kép csak valamivel több, mint 600 KiB lemezterületet foglal el.

A `'%Y-%m-%d-%H-%M-%S.jpg'` paraméter beállítja a scrot által mentett képek fájlnevét. Az `%Y` az évet, a `%m` a hónapot, a `%d` a hónap napját jelöli. A `%H` az órát, a `%M` a percet, a `%S` pedig a másodperceket jelenti. Végül megmondjuk a scrot-nak, hogy a képernyőképeket JPG-fájlként mentse. Ellenkező esetben a scrot alapértelmezés szerint PNG-fájlokat használ, amelyek jóval nagyobbak, mint a JPG-fájlok.

A `for ... do` ciklus 30 másodpercenként egyszer fut le, amíg el nem készül 10 képernyőkép. Tehát van egy ötperces ablakom a képernyőképeim rögzítésére, 30 másodperces időközönként egyenletesen elosztva. Eddig ez tökéletesen megfelelt az igényeimnek (amit korábban kifejtettem).



a szkript befejezéséhez egy másik Zenity párbeszédpanel jelenik meg mind a 10 képernyőkép rögzítése után, értesítve a felhasználót, hogy a szkript befejeződött. És akkor a szkript kilép.

A szkript futtatása

Korábban említettem, a szkript tökéletesen megfelel az igényeimnek. De bash szkript lévén, te is módosíthatod, hogy megfeleljen az igényeidnek. Ami a parancssori paramétereket illeti, a szkriptnek nem kell semmi.

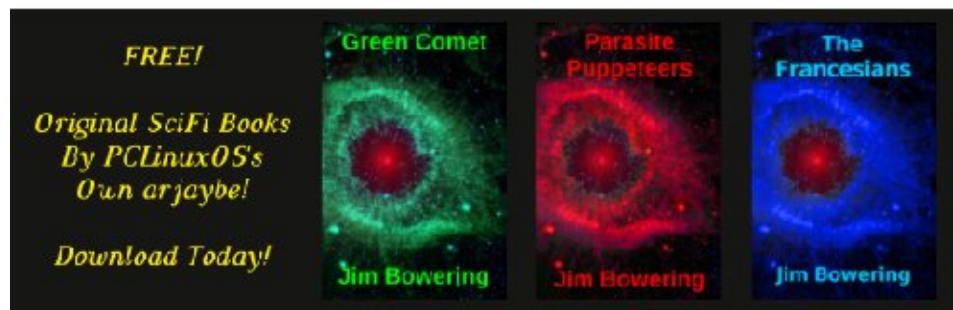
A szkriptet tárold ott, ahol általában menteni szoktad az összes felhasználói szintű szkriptet. A `/home` könyvtárban van egy `/bin` nevű könyvtár, ami a rendszerem `$PATH` utasításában is meg van határozva. Tehát elindíthatja a szkriptet egy terminálból, vagy létrehozhat egy indítót a panelen, ami egyetlen egérgattintásra elindítja a szkriptet.

A legtöbb webináriumom egészkor kezdődik. A résztvevők legfeljebb 15 perccel a tervezett kezdési időpont előtt jelentkezhetnek be a webináriumra. Tehát, ha beírom az „1”-et az óra párbeszédpanelbe, és a 10

percet a perc párbeszédablakba, rögzíteni tudom a kódot a webinárium végén. Ez lehetővé teszi, hogy a szkriptem 30 másodperces időközönként képernyőképeket rögzítsen a webinárium végén, a webinárium utolsó öt percében. Ez nagymértékben javítja az esélyeimet a „mindenható kód” rögzítésére, szemben az egyszerű „vaktában kattintással” bármi más képernyőképes segédprogrammal.

A szkriptet elég könnyű átírni, ha kell. Tehát, ha akarnám, a szkriptben a `sleep 30s` parancsot (amely 30 másodpercenkénti képernyőkép indításáért felelős) lecserélhetem `60s` alvásra. Ez az egyszerű változtatás azt eredményezi, hogy a szkript percenként egy képernyőképet készít 10 percig. Ha ezt tenném, a perc párbeszédpanelen beírt percek 5 percre módosítanám, aminek következtében a forgatókönyv percenként egy képernyőképet készítené a webinárium utolsó 10 percében.

Egy másik rendkívül könnyen végrehajtható változtatás a scrot által készített képernyőképek minőségének megváltoztatása. Valójában két beállítási lehetőség van. Az első az, hogy módosítsa a „`--quality 10`”-et „`--quality 90`”-re, ha nagyobb felbontású képet szeretne (természetesen tetszőleges számot használhat, amire szüksége van... Én csak a „90”-et használtam példaként). A legjobb, ha a „minőség” utáni számot százalékban gondolja. A nagyobb számok alacsonyabb tömörítést és nagyobb fájlméretet, míg az alacsonyabb számok nagyobb tömörítést és kisebb fájlméretet eredményeznek. A scrot második megváltoztatható eleme a JPG képfájlok helyett PNG képfájlokba mentés, a fájlnev definíciójának végén egyszerűen a fájl kiterjesztésének megváltoztatásával. A PNG fájlok eredendően élesebbek (nagyobb felbontásúak), mint a JPG fájlok, egyszerűen azért, mert a PNG fájlok veszteségmentes tömörítést használnak, míg a JPG fájlok veszteséges tömörítést.



Összegzés

Eredetileg ezt a praktikus szkriptet egy konkrét igényem kielégítésére készítettem. Ugyanakkor, egészen biztos vagyok benne, hogy szükség esetén sok más olyan elvárások esetén is megfelel, amikor képernyőképeket kell készítenie valamikor a jövőben.

Most végre kihúzhatom a késleltetett képernyőkép segédprogram iránti igényemet a „kívánságlistámról”.



A fenti kép az egyik tesztképernyőképemből vett mintarészlet, 10-es minőségi szinttel. Amint látható, elég jól olvasható, de közel sem áll ahhoz, amit élesnek vagy nagy felbontásnak neveznék.

Szórakozásból, illetve a szkript tesztelésére lefekvés előtt 8 óra 0 percre állítottam az időzítőt. Ezután lecsuktam a laptopom fedelét, és lefeküdtem. Amikor másnap reggel felébredtem, megtaláltam a szkriptet, ahogy kötelességtudóan elkészítette a képernyőképeket, az utasításoknak megfelelően. A tesztelés során 1 perces, 2 perces, 5 perces, 20 perces és egy órás késleltetéssel is próbálkoztam. Minden pontosan úgy működött, ahogy reméltem, és ahogy terveztem.

A szkript [letölthető](#) (a fentebb bemutatott formájában... ez egy meglehetősen kicsi, 1,2 KiB-os fájl) a magazin webhelyéről. A szkript neve „delayed-screenshot.sh.txt”. Mentse el a szkriptet oda, ahol általában menti a szkripteket, távolítsa el a „.txt” fájlkiterjesztést, és tegye futtathatóvá a fájlt.

