

Több parancssori ablak GNU Screen-nel

PCLinuxOS Magazine – 2015. október

Írta: Peter Kelly (critter)

Amikor KDE-vel, vagy a kedvenc PCLinuxOS grafikus környezetben dolgozol, legyen az bármi más, az egyes alkalmazások saját ablakukban nyílnak meg. Ez lehetővé teszi, hogy ugrálj az alkalmazások között, nyitva hagyva a munkádat az előzőleg használt ablakban addig, amíg vissza nem lépsz oda, mented és bezárod az ablakot. Módodban áll, hogy másik virtuális munkaterületre lépj és további alkalmazásokat, akár másikat, akár a már megnyitottal azonos alkalmazásokat nyiss meg. A KDE még ennél is messzebb megy az aktivitások bevezetésével.

Ez a szabadság akkor nagyon hasznos, amikor más információkat is kell használnod, zenét akarsz váltani, vagy bejövő üzenetre válaszolni. Nagy mértékű rugalmasságot biztosít. Sajnos, ez a szabadság parancssori környezetben végzett munka közben nem áll rendelkezésre, vagy mégis?

Természetesen igen. A Unix és a Linux parancssorra alapozó operációs rendszer, még ha a modernebb kiadások grafikus kezelői felületen is futnak. A terminál alapú eredet következtében az igény a rugalmasságra már évtizedekkel ezelőtt jelentkezett és egy megoldás, a screen, 1987 óta van.

Számos megoldás van erre. Két eszköz található meg a PCLinuxOS tárolóiban, a screen, ami a GNU projekt része és a tmux, amit pedig BSD licenc alatt adtak ki. A byobu, ami a GNU screen egy kezelői felülete és a tmux is érdekesek, de az előbbi a PCLinuxOS tárolóiban jelenleg nem érhető el.

A screen és a tmux hasonló jellegű terminál emulátor, de a screen a „szabványos”, elérhető majd minden Linux disztribúciónál, ezért én itt a screen-re koncentrálok. Ha a cikk elolvasása után úgy döntesz, hogy használnod némely megoldását, akkor próbáld ki a tmux-ot is. Lehet, hogy jobban fog tetszeni, vagy lesznek olyan tulajdonságai, amik a screen alatt nem érhetőek el.

A screen nem egy újabb terminál emulátor, hanem egy ablakkezelő a parancssor számára (nevezik még terminál multiplexernek is), lehetővé téve, hogy egyetlen terminálban több ablak legyen nyitva, virtuális terminál (akár mint a CTRL + F1-F6 lenyomásával), vagy terminál emulátor. Kimondottan úgy mozoghatsz bennük, mint a KDE-ben. Mindegyik a terminál keretein belül marad, de váltogathatod azokat, hozzáadhatsz és törölhetsz ablakot, illetve a könnyebb hivatkozás érdekében elnevezheted. A munkamenet (ablakcsoport) elnevezhető a későbbi

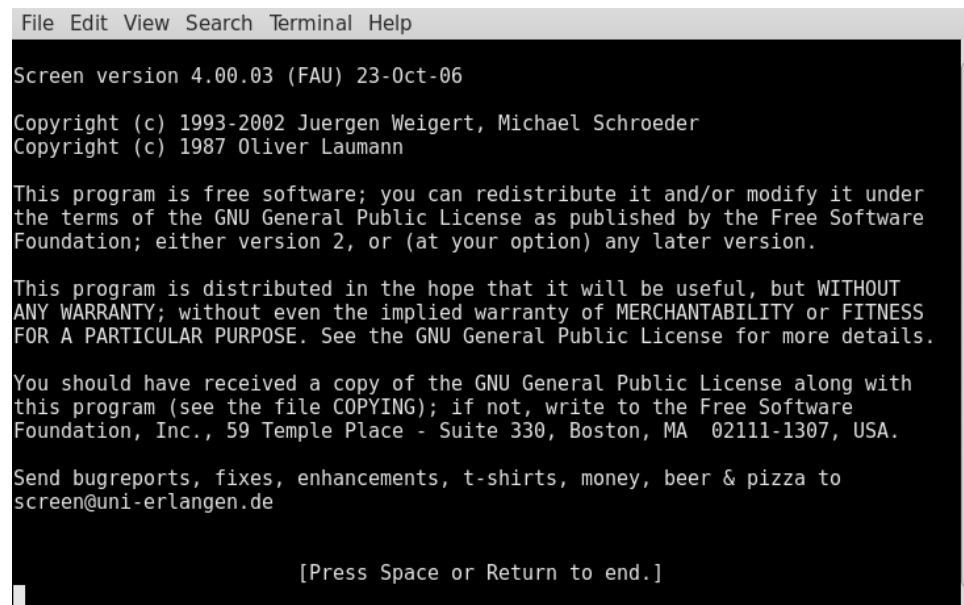
előhíváshoz. Egy gépen több screen is nyitva lehet egyszerre, ha kell szétválaszthatod és újra csatlakozhatod. Azt is lehet, hogy egy gépen indítasz egy screen munkamenetet, leválasztod, majd bejelentkezel és egy másik gépen újra csatlakozol, hogy folytasd a munkádat.

Kezdeti lépések

Azután, hogy a PCLinuxOS tárolóiból telepítetted a screen-t, a következő parancsot kell futtatnod egyszerű felhasználóként, a saját home könyvtáradból, hogy ne kapj ehhez hasonló figyelmeztetést.
“Directory /home/pete/tmp must have mode 700.”

chmod 700 ~/tmp

Amikor a screen-t először indítod el a „screen” beírásával terminálba, vagy terminál emulátor ablakába, üdvözlő képernyő fogad, ami kéri a szóköz billentyű lenyomását a folytatáshoz.



```
File Edit View Search Terminal Help
Screen version 4.00.03 (FAU) 23-Oct-06
Copyright (c) 1993-2002 Juergen Weigert, Michael Schroeder
Copyright (c) 1987 Oliver Laumann

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software
Foundation; either version 2, or (at your option) any later version.

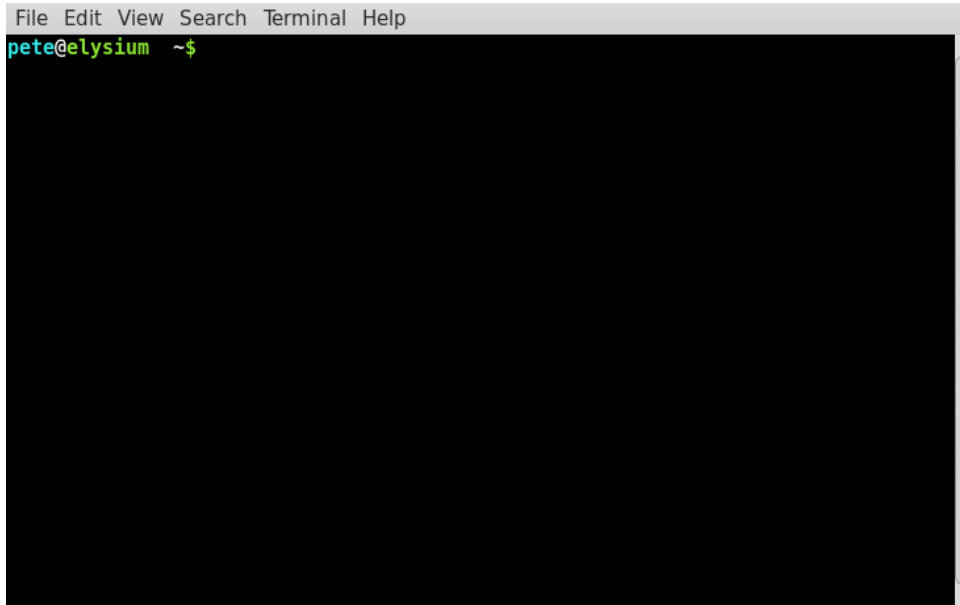
This program is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with
this program (see the file COPYING); if not, write to the Free Software
Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Send bugreports, fixes, enhancements, t-shirts, money, beer & pizza to
screen@uni-erlangen.de

[Press Space or Return to end.]
```

Amikor megtetted, úgy tűnik, mintha mi sem történt volna és a program futása megszakadt, noha a screen rendben fut. Egyszerűen csak eltűnt előled, ahogy azt egy jól nevelt alkalmazás teszi.



```
File Edit View Search Terminal Help
pete@elysium ~$
```

A tökéletes integráció érdekében az alapbeállítás szerinti viselkedés megváltoztatható, miközben továbbra is meg tudjuk mondani, hogy hol vagyunk.

A screen eszköz, a többi linuxos dologhoz hasonlóan, egyszerű szöveges fájlból veszi a beállításait. Az alap beállító fájl a /etc/screerc alatt található, ami azon felhasználók számára érvényes, akiknek nincs személyes beállító fájljuk. A screen először egy rejtett, .screenrc nevű fájlt keres a home könyvtárban. Ha az létezik, akkor az általános beállító fájl helyett azt veszi elő. Így lehet pontosan úgy beállított screen verziód, amelyet szeretnél, ami sokkal jobb dolog.

Sajnos, a legtöbb Linuxos dologhoz hasonlóan, noha a beállító fájl sima szöveges, a tartalma nem teljesen az, és kell egy kis idő mire megértheted a tartalmát. A dokumentáció mindent leír, de azért én bemutatom, hogyan állítottam be a saját .screenrc fájloamat, amit innen leszedhetsz.

Először, amit tennünk kell, hogy megszabadulunk az üdvözlő képernyőtől. Ehhez, nyiss egy sima szövegszerkesztőt és írd be ezt a sort:

```
startup_message off
```

Mentsd a fájlt a home könyvtárba .screenrc név alatt (a pont legyen az elején). A következő, hogy legyen a szokásostól eltérő parancsör (prompt), amikor a screen-t használjuk. Én a következő sorokat raktam be a ~/.bashrc fájlomba a két eltérő prompt leírásaként:

```
if [ -z $STY ]
then

PS1='[\033[1;36m]\u\[\033[1;37m]@\[\033[1;32m]\h \
\[\033[1;37m] \w\[\033[1;32m]\$[\033[0;0m] '
else

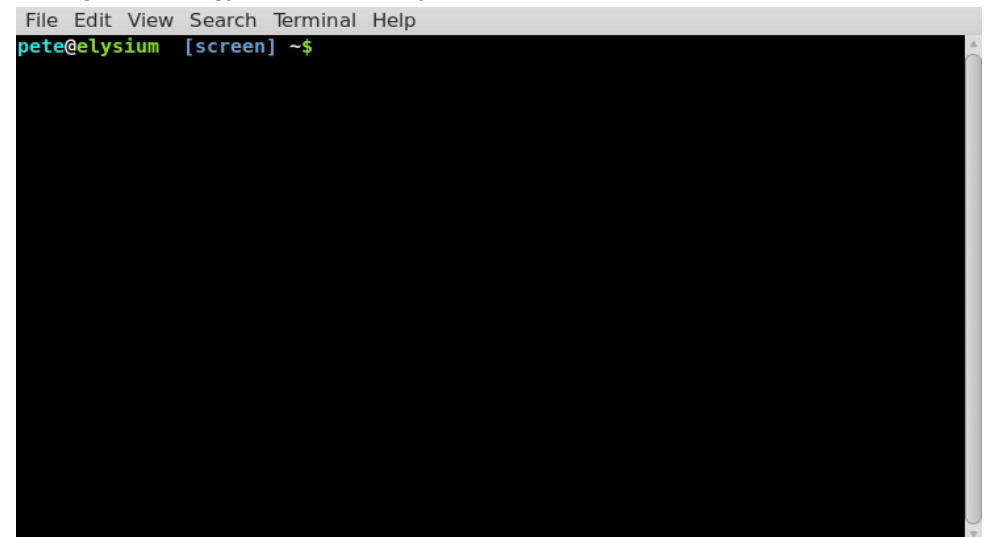
PS1='[\033[1;36m]\u\[\033[1;37m]@\[\033[1;32m]\h \
\[\033[1;37m] \w\[\033[1;34m][screen]\[\033[1;37m]
\w\[\033[1;32m]\$ \
\[\033[0;0m] '
fi
```

Ezt követi ez a sor, hogy a megfelelő prompt kerüljön a rendszerváltozóba:

```
export PS1
```

A promptnak nem kell feltétlenül ilyen viccesnek lennie, de legyenek eltérőek. Ha prompt-okról több információ kellene, akkor ennek az oldalnak a végét nézd át <http://pclosmag.com/html/Issues/200911/page19.html>.

Az eredmény a következő prompt-ot eredményezi a screen-be belépéskor, kétséget sem hagyva arról, merre jársz.



```
File Edit View Search Terminal Help
pete@elysium [screen] ~$
```

Mielőtt további finomításba kezdenénk, lássunk néhány olyan dolgot, amit a screen-nel tudunk kezdeni. Átnézünk néhány hasznosabb parancsot, és a cikk végén egy puska formájában összefoglalót készítek.

A parancsok

A screen utasításához először a „parancs karaktert” kell beírni, ami alapbeállítás szerint <CTRL>+a (a továbbiakban **C-a**), ezt egy karakterből, vagy billentyű lenyomásából álló parancs követi. A parancs karakter biztosítja, hogy a parancs a screen-hez kerüljön és ne a screen-ben futtatott folyamathoz. Például a screen bezárásához gépelj **C-a l**-t.

Segítségképpen a screen értelmezi azt is, amikor lenyomott billentyűt követi a Control és a billentyű lenyomása, arra az esetre, ha a elfelejtenéd levenni az ujjadat a Control billentyűről.

Az alpból beállított **C-a** megváltoztatható, aminek azok örülnek, akik ezt a billentyűkombinációt arra használják, hogy a kurzort a parancssor elejére vigyék (mint én), de a „**C-a a**” írott „**C-a**” küld a screen-nek, ami szintén működik.

Az elérhető parancsok a „C-a ?” lenyomására megtekinthetők.

A screen néhány tulajdonságának bemutatásához indítsd el, majd írd be

top

```
File Edit View Search Terminal Help
Num Name                               Flags
0 bash                                  $
1 bash                                  $
2 bash                                  $
```

Egy megszakítás nélkül futó folyamatot látunk a terminálban, ami betölti a terminálablakot. Ezután add ki a „**C-a c**” parancsot. Ez új, üres terminálablakot hoz létre, amire átvált. Írj be egy másik parancsot. Az „mc” jó lesz, ha a midnight commander telepítve van nálad (ha nem lenne, akkor telepítsd, nem fogod megbánni). Ismét írd be a „**C-a c**”-t egy újabb ablakot létrehozva, majd írd be **C-a** „”, az ablakban pillanatnyilag elérhető ablakok kilistázása érdekében.

A nyíl billentyűkkel mozogva válaszd ki a nullás ablakot, üss Enter-t és ismét a top-ot bámulhatod, a legelső folyamatot, amit az első screen ablakban indítottunk el.

A gond ezzel a listázással az, hogy mindhárom ablak neve bash, ami természetesen az egyes ablakokban futó folyamat. De hogy melyik ablakban éppen mi folyik? Ezen lehet javítani azzal, hogy sokkal valóságosabb nevet adunk az ablakoknak. Gépelj „**C-a A**”-t (egy nagybetűs A), ami balra lent megjelenít egy panelt, ahol az ablakokat átnevezheted. Visszafelé töröld a bash nevet és írd be, hogy top.

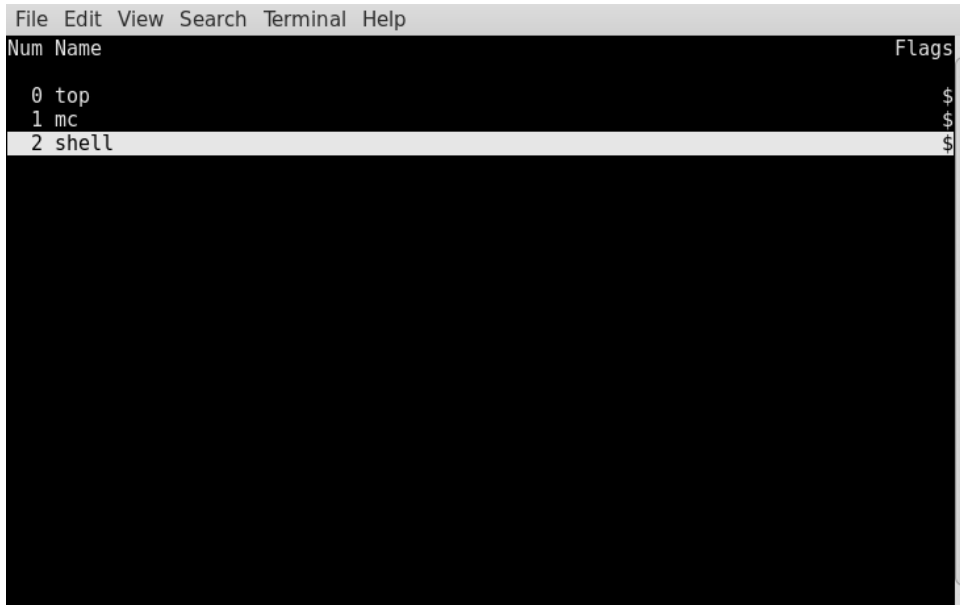
```
File Edit View Search Terminal Help
top - 19:58:05 up 2:20, 4 users, load average: 0.00, 0.01, 0.05
Tasks: 131 total, 2 running, 128 sleeping, 0 stopped, 1 zombie
Cpu(s): 0.7%us, 3.7%sy, 0.0%ni, 94.6%id, 1.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1023844k total, 520068k used, 503776k free, 100672k buffers
Swap: 971896k total, 0k used, 971896k free, 185548k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
912	root	20	0	321m	39m	18m	S	0.7	3.9	0:51.40	X
2393	pete	20	0	301m	58m	21m	S	0.7	5.9	0:50.50	net applet
1	root	20	0	4276	1504	1400	S	0.0	0.1	0:01.41	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.57	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
7	root	-3	0	0	0	0	S	0.0	0.0	0:00.47	rcu_sched
8	root	-3	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	RT	0	0	0	0	S	0.0	0.0	0:00.03	migration/0
10	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	khelper
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
12	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
13	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	perf
14	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	writeback
15	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
16	root	39	19	0	0	0	S	0.0	0.0	0:00.00	khugepaged

Set window's title to: **top**

C-a n, vagy **C-a** **szóköz** lenyomásával léphetsz a következő ablakra (**C-a p**, vagy **C-a** **törlés vissza** léptet az előző ablakokra, ciklikusan) és nevezd át mc-re. Ismételd meg az utolsó ablakkal is a folyamatot és nevezd át shell-nek, mivel az új ablakok bash néven fognak futni. Most ismét írd be **C-a** „” és megtekintheted a változtatásokat, így mindjárt más.

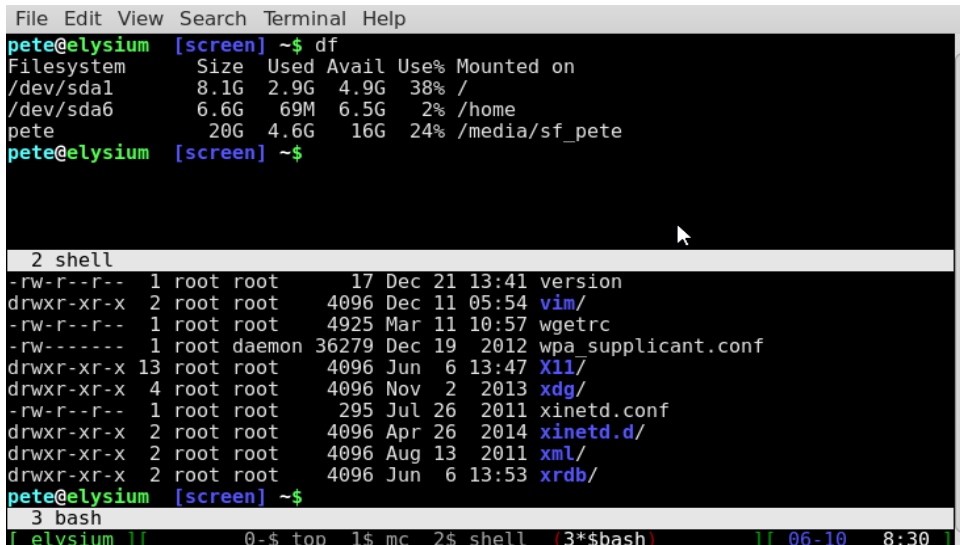
Több parancssori ablak GNU Screen-nel



```
File Edit View Search Terminal Help
Num Name                               Flags
0 top
1 mc
2 shell
```

Válaszd ki a második (shell nevű) ablakot és írd „C-a k”-t. Azt írja ki, hogy “Really kill this window [y/n]” Answer “y” (valóban bezárod az ablakot?). Válasz: „y”. Erre törli az ablakot és a következő kisebb sorszámú ablakra lép, esetünkben az 1-re, ami „mc” nevet viseli.

Egy adott ablakra ugráshoz használd a „C-a '”-t és balra lent kérni fogja a kívánt ablak megjelölését. Használhatod az ablak sorszámát, vagy a nevét. Ha az



```
File Edit View Search Terminal Help
pete@elysium [screen] ~$ df
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       8.1G  2.9G  4.9G  38% /
/dev/sda6       6.6G   69M  6.5G   2% /home
pete            20G   4.6G  16G   24% /media/sf_pete
pete@elysium [screen] ~$

2 shell
-rw-r--r-- 1 root root    17 Dec 21 13:41 version
drwxr-xr-x 2 root root  4096 Dec 11 05:54 vim/
-rw-r--r-- 1 root root  4925 Mar 11 10:57 wgetrc
-rw----- 1 root daemon 36279 Dec 19 2012 wpa_supplicant.conf
drwxr-xr-x 13 root root  4096 Jun  6 13:47 X11/
drwxr-xr-x 4 root root  4096 Nov  2 2013 xdg/
-rw-r--r-- 1 root root   295 Jul 26 2011 xinetd.conf
drwxr-xr-x 2 root root  4096 Apr 26 2014 xinetd.d/
drwxr-xr-x 2 root root  4096 Aug 13 2011 xml/
drwxr-xr-x 2 root root  4096 Jun  6 13:53 xrdp/
pete@elysium [screen] ~$

3 bash
[ elysium ]| 0-$ top 1$ mc 2$ shell (3*$bash) |[ 06-10 8:30 ]
```

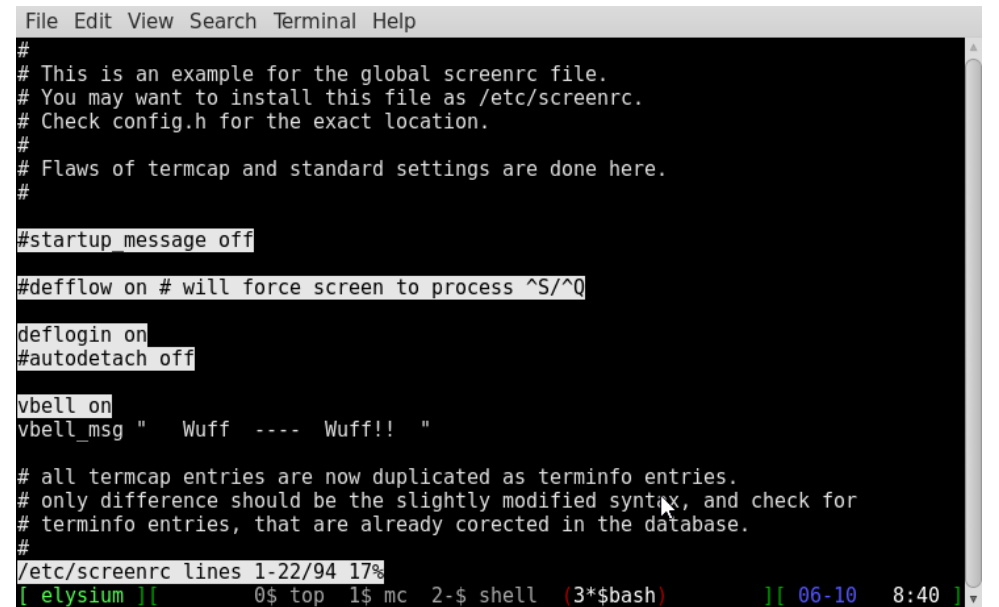
aktuális oldalt határozza meg, akkor csak tömören annyit mond „This is window 1 (mc).” (ez az1-es (mc) ablak). A 0-9 számú ablakokhoz a **C-a {sorszám}** is jó. Az előző ablak a „C-a C-a”-val visszahozható.

A felosztás (split) paranccsal „C-a S” (nagy S) az aktuális ablak függőlegesen ketté osztható. Az új ablak üres és semmi nem fut benne. Az új ablakra a „C-a <Tabulátor>”-ral léphetsz be, ahol „C-a c”-vel indíthatsz új folyamatot. A felosztás törlése „C-a X”-szel (nagy X) történhet, amikor benne vagy.

A screen lehetővé teszi a másolás-beillesztést ablakok között a következő parancsokkal

„C-a [” és „C-a]”.

A „C-a [” parancs kiadásával teljes képernyős másolás módba kerülsz. Itt a nyíl billentyűkkel lehet mozogni. A 0 billentyű visz a sor elejére és a \$ pedig a sor végére. Kereshetsz szöveget a / jellel előre és a ? jellel visszafelé. Egy hosszabb anyagban lapozásra a <Ctrl>+b és a <Ctrl>+f billentyűk használhatók. Amikor eléred annak a szövegrésznek az elejét, amit másolni akarsz, akkor nyomj <Enter>-t és menj a másolandó rész végére. Mozgás közben a szöveget kijelöli és végül nyomd meg az <Enter>-t ismét. Ezzel a kijelölt szöveg a másolásra használt pufferbe kerül és a képernyő alján kijelzi, hogy mennyi szöveget másolt be.



```
File Edit View Search Terminal Help
#
# This is an example for the global screenrc file.
# You may want to install this file as /etc/screenrc.
# Check config.h for the exact location.
#
# Flaws of termcap and standard settings are done here.
#
#startup message off
#defflow on # will force screen to process ^S/^Q
deflogin on
#autodetach off

vbell on
vbell_msg " Wuff ---- Wuff!! "

# all termcap entries are now duplicated as terminfo entries.
# only difference should be the slightly modified syntax, and check for
# terminfo entries, that are already corrected in the database.
#
/etc/screenrc lines 1-22/94 17%
[ elysium ]| 0-$ top 1$ mc 2-$ shell (3*$bash) |[ 06-10 8:40 ]
```

Most nyisd meg a kedvenc szövegszerkesztőt, vagy egy másik ablakot és írd be „C-a]”. A szöveget bemásolja a szerkesztőbe.



Több parancssori ablak GNU Screen-nel

Úgy tűnhet, mintha a top és a Midnight Commander elszállt volna, de valójában továbbra is elérhetőek a screen-ben. Gépelj be <Ctrl>+F1-et, hogy virtuális terminálba kerülj (<Ctrl>+F8 visszavisz a grafikus környezetbe, ha honvágyad lenne). (Ford.: az F billentyűk sorszáma disztró függő.)

Most írd be, hogy **screen -r** és két képernyőd lesz a „top” és az „mc”.

Ugyanezt más gépeken is megcsinálhatod. Ha van hozzáféréseid más géphez a hálózaton keresztül, akkor bejelentkezhetsz és kapcsolódhatsz egy korábban leválasztott screen folyamathoz és dolgozhatsz tovább.

```
File Edit View Search Terminal Help
GNU nano 2.4.1 New Buffer Modified
#startup_message off
#defflow on # will force screen to process ^S/^Q
deflogin on
#autodetach off
vbell on

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
[ elysium ] [ 0$ top 1$ mc (2*$shell) 3-$ bash 4$ bash ] [ 06-10 9:05 ]
```

Ne másold be a szöveget közvetlenül parancssori ablakba, mivel a héj megpróbálja végrehajtani az egyes sorokat, ami előre megjósolhatatlan és potenciálisan katasztrofális eredményt hozhat.

Most jön az okosság. Gépelj „**C-a d**”-t és kikerülsz a képernyőről, visszakapod a parancssort, ami eltérő lesz attól, amiben előzőleg voltunk a screen-ben és egy egyszerű „[detached]” (leválasztott) feliratot kap.

```
File Edit View Search Terminal Help
pete@elysium ~$ screen
[detached]
pete@elysium ~$
```

```
File Edit View Search Terminal Help
pete@elysium wd-> ~
$ ssh pete@asgard
Last login: Mon Jun 8 14:06:32 2015 from asgard
pete@asgard wd-> ~
$ screen -r
```

Itt láthatod, ahogy a hálózaton lévő másik, asgard nevű géphez csatlakozom és átveszek egy screen futást, amit korábban indítottam.

```
File Edit View Search Terminal Help
pete@asgard wd->[screen] ~
$ df
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/sda5       ext4   10G   3.2G   6.3G  34% /
/dev/sda8       ext4    14G   794M   14G   6% /home
/dev/sda3       ext4   75G   18G   54G  25% /home/pete/data
pete@asgard wd->[screen] ~
$

14:08 Monday 08 Jun 2015 [0 bash]
```

Ez akkor jó, ha távoli kiszolgálón dolgozol, vagy egy sor parancssori eszközöd van megnyitva egy hosszabb ideje futó gépen. Ha távolról dolgozol és hirtelen elveszíted a kapcsolatot, egyszerűen újra kapcsolódsz és újra csatlakozol a screen folyamatodhoz. Be- és kijelentkezhetsz, de addig, ameddig nem

kapcsolod ki újraindításhoz a gazdagépet, amin a folyamatot indító screen fut, az a folyamat elérhető marad az adott screen-ben.

Ha egynél több screen-t akarsz futtatni, elnevezheted a folyamatot a screen indításakor a következő paranccsal

screen -S {session-name}

Így könnyebb az egyes screen-folyamatok között kapcsolgatni. Név nélkül, a PID-et kellene használni. A pillanatnyilag futó screen folyamatok listázása

screen -ls

```
File Edit View Search Terminal Help
pete@elysium ~$ screen -ls
There are screens on:
      8891.pts-0.elysium      (Detached)
      8862.pts-0.elysium      (Detached)
2 Sockets in /home/pete/tmp.

pete@elysium ~$
```

A képernyőkép két folyamatot mutat, pillanatnyilag mindkettő leválasztva, de az egyikük újra csatolásához be kell írni a teljes nevet

screen -r 8891.pts-0.elysium

Ez elég unalmas lehet. Amikor folyamatot elnevezed, valami ilyesmit kapsz:

```
File Edit View Search Terminal Help
pete@elysium ~$ screen -ls
There are screens on:
      9683.coding      (Detached)
      9645.logs        (Detached)
2 Sockets in /home/pete/tmp.

pete@elysium ~$
```

Most két nevesített folyamatom van, így a rendszernaplók ellenőrzéséhez használhatom

screen -r logs

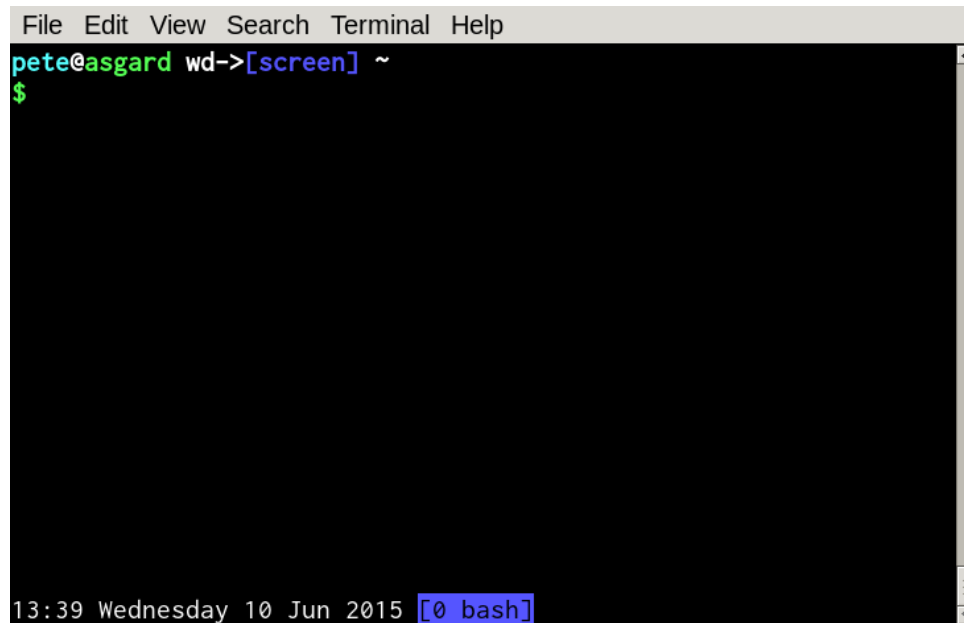
Ez így sokkal egyszerűbb. A folyamatok neve legyen egyedi. Ha pl. „test1” és „test2” screen neveket használsz, nem tudsz különbséget tenni. Talán mégis lehet, de nem találtam meg.

Ha csatlakozni akarsz egy már csatolt folyamathoz, pl. egy távoli rendszerről, akkor kombinálnod kell, leválasztani és újracsatolni parancsokkal, valahogy így.

```
File Edit View Search Terminal Help
pete@elysium wd-> ~
$ ssh pete@asgard
Last login: Wed Jun 10 13:36:39 2015 from elysium
pete@asgard wd-> ~
$ screen -ls
There is a screen on:
      11068.pts-0.asgard      (Attached)
1 Socket in /home/pete/tmp.

pete@asgard wd-> ~
$ screen -dr 11068.pts-0.asgard
```

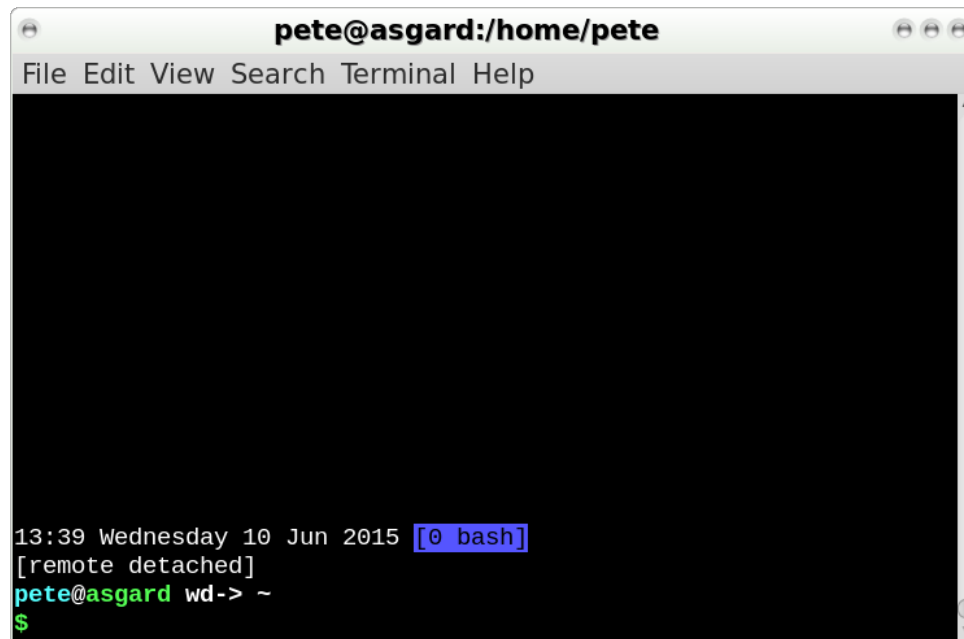
Ezzel beléptél a folyamatba ...



```
File Edit View Search Terminal Help
pete@asgard wd->[screen] ~
$

13:39 Wednesday 10 Jun 2015 [0 bash]
```

A távoli terminál valahogy így néz majd ki.



```
pete@asgard:/home/pete
File Edit View Search Terminal Help

13:39 Wednesday 10 Jun 2015 [0 bash]
[remote detached]
pete@asgard wd-> ~
$
```

A screen saját parancssori prompt-tal bír, amit „**C-a** :”-val érhetsz el. A prompt kettőspont formájú és az ablakban balra lent jelenik meg, ahová majd beírod a parancsot. A kiadható parancsok leírása a haladó felhasználók számára a dokumentációban megtalálható, de a legtöbb parancsot a `~/screenrc`, vagy `/etc/screenrc` tartalmazza. Az a három parancs, amit a screen-ben esetleg futtathatsz a következő:

C-a : quit – bezárja az összes ablakot és kilép a screen-futásból. Ez azt jelenti, hogy nem lehet újracsatolni.

C-a : source ~/screenrc – újratölti a konfigurációs fájlot, vagy új konfigurációs fájlt tölt be, amennyiben meghatározzuk.

C-a : sessionname név – elnevezi, vagy átnevezi az aktuális screen-folyamatot.

Ha egy screen képernyőképét nézed egy távoli gépen, láthatsz egy státuszsort alul, amink jelen esetben nincs. Ez a `~/screenrc` fájlban hozható létre néhány beállítással, amiket most átnézünk.

Screen testreszabása

A státuszsor a képernyő alján lehet egyszerű és bonyolult is, ahogy akard. Ugyanakkor, a sor beállítása nem ördögösség és a kinézete függ az általad használt termináltól is. Noha a kód magyarázata a dokumentációban megtalálható, mégis elég homályos, ezért csaltam, és egy a neten találtat másoltam ki, amit azután addig „piszkáltam”, amíg a kívánt eredményt nem kaptam. Amit használtam, azt a arch wiki-jén találtam (nagyon hasznos forrás). Elég sokat változtattam rajta, de működik mate-terminal-on, konsole-on, xterm-en és az ssh kapcsolaton is. Ezt a két sort raktam hozzá a `.screenrc`-mhez.

```
caption string "%{= kW}%c %LD %d %M %Y %{= kB}%{+s}[%n %t]%= kW)"
caption always
```

Megjeleníti az aktuális időt és dátumot brit formában és az aktuális screen-t kézzel emeli ki. A karaktorsor makrók sorozatából, vagy „escape láncból”, attribútumokból, módosítókból és betűkből áll össze. Mondtam neked, hogy nem egyszerű. Az első sor így működik:

`%{= kW}` - a „%” egy escape szekvencia indítását jelzi és a kapcsos zárójelek fogják közre az elemeket. Az „=” jelenti az aktuális attribútum átállítását az azt követőre és a „+” jelezni az attribútum hozzáadását a létező attribútumokhoz. A „kW” páros a háttér-, előtérszín leírása ebben a sorrendben. A „k” jelenti a fekete hátteret, a „W” pedig a fehér előteret (white), ám a nagybetű miatt az fényes lesz.

%c – az escape karakter hivatkozik az aktuális időre HH:MM (óra-perc) 24 órás formátumban. Ha 12 órás formát akarsz, akkor a „c” legyen nagybetűs.

%LD – a megelőző szóköz számít és ténylegesen beilleszt egy szóközt. A nap lehet szám „%d”, név „%D”, vagy mint itt az „L” miatt a névvel együtt „%LD”.

%d – újabb szóköz és utána a nap számformában „%d”.

%M – szóköz megint és a hónap rövidítve. A”%m” számot ad vissza, amíg a „%LM” a hónap hosszú, vagy teljes nevét adja.

%Y – mindkét oldalánál szóköz és a „%Y” négykarakteres, a „%y” kétkarakteres évszámot jelenti.

%{= kB}%{+s} – hasonlít az első kifejezéshez, fekete háttér fényes kék előtérrel. Ez alkalommal közvetlenül követi egy „%{+s}”, ami hozzáadja a kiemelés (standout) változót, fekete fényes kék előtt.

[%n %t] – a a szögletes zárójeleket és a szóközt karakterként kezeli. A „%n” adja ki a screen számát és a „%t” a screen nevét.

%{= kW} – végül visszaállítja a fényes fehérre fekete háttér előttire.

Ez nem túlzottan bonyolult, de sok munkának tűnik ahhoz, hogy megjelenítsünk és megformáljunk némi egyszerű információt.

A második sor egyszerűen közli a screen-nel, hogy mindig jelenítse meg a sort.

Más szövegeket, mint a „**C-a w**”-re kiadott ablaklista, időszakosan felül lesz írva.

Ha a termináloed támogatja a 256 színt és kihasználnád ezt, akkor add ezt a sort hozzá.

term xterm-256color

Gyakran, ha olyan eszközt használasz, mint a vi, vagy a nano szövegszerkesztő, a szöveg a képernyőn maradhat az eszközből kilépés után. Ennek kiküszöbölésére add hozzá ezt a sort.

altscreen on

Egyes terminálok villognak csengőt imitáló, egy jellemző, ami az írógépek idejéből maradt meg. Ennek az idegesítő szokásnak a kikapcsolásához add hozzá ezt a sort.

GNU Screen Command Reference

The control key, control a (**C-a**) by default, must be pressed before any of these commands.

?	Show a list of available commands. Space will show more if they don't all fit and return will clear the list and take you back.
c	Create a new window.
"	Show a selectable list of available windows.
A	Rename a window.
n	Go to the next window.
space	As above.
p	Go to the previous window.
backspace	As above.
0-9	Go to the numbered window.
C-a	Go to the previously open window (That's control-a twice).
'	Prompt for the window name or number to switch to.
S	Create a horizontal split in the terminal window. May be repeated.
Tab	Switch focus to the next region of a split screen.
a	send a literal Control-a to the screen.
C	Clear the current screen.
h	Send a hard copy of the current screen to the file <u>hardcopy_n</u> where n is the number of the screen.
H	Start/stop logging of the current screen to the file <u>screenlog_n</u> where n is the number of the screen.
N	Show the number of the current window.
\	Quit screen – kills all windows.
x	Lock the terminal. Requires password to unlock.
X	Kill this region.
Q	Kill all regions but this one.
k	Kill the current window.
z	Suspend the screen session. Type <u>f g</u> to resume as normal bash job control.
d	Detach screen from this terminal.
[Start a copy session.
]	Copy the contents of the copy buffer to the current screen.

vbell off

A „C-a” -ra megjelenített ablaklista formátuma beállítható a felirat szövegéhez hasonló eljárás szerint.

```
windowlist string "%3n %={ kG}%t"
```

A „%3n” jeleníti meg a screen sorszámát három karakter szélességben elosztva. A „%={ kG}%t” jeleníti meg screen nevét fényes zöldben fekete háttér előtt.

A screen néhány előre meghatározott ablakkal való indításához

```
screen -t top 1 top
screen -t mc 2 mc
screen -t shell 3 bash
```

A fenti formátum

```
screen -t {neve} {sorszáma} {parancs}
```

Végül létrehozhatod a saját billentyűparancsaidat.

```
bind m screen -t 'mc' 8 mc
```

Ez lehetővé teszi, hogy a „C-a m” a Midnight Commander-t a 8-as ablakban nyissa meg, nevezze el mc-nek, majd váltson át rá. Ha a 8-as ablak már használatban van, akkor a soron következő magasabb számot veszi.

Ez a bemutató csak egy igen kis minta a screen eszköz képességeiről, de amiket itt bemutattam, azok az általam leghasznosabbnak talált tulajdonságai.



The PCLinuxOS Magazine Special Editions!

Get Your Free Copies Today!