

Quelques scripts (f)utiles

- Objet : liste en vrac de scripts plus ou moins utiles ... donc indispensables :D
- Niveau requis :
[débutant](#), [avisé](#)
- Commentaires : *page à piller*.
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊



Les scripts sont des fichiers texte rédigés dans différents langages ([bash](#), python, perl, ruby...) et qui envoient des commandes à votre système.

l'intérêt majeur du script est de regrouper en un appel, une série de commandes et ainsi, d'effectuer des tâches complexes simplement.

Pour être actif, un script doit être **exécutable** ([chmod](#) +x)

chrono xxs

un chronomètre graphique rudimentaire mais efficace ;) [sources by thuban](#)

[chrono-xxs](#)

```
#!/bin/sh
#
WAIT=`zenity --entry --title="Timer" \
    --text="Temps de l'activité? (ex: 10m)"`

if [ -n "$(echo $WAIT | grep -o 'm')" ]; then
    # minutes
    FACT=60
    WAIT=$(echo $WAIT | sed 's;m;;')
elif [ -n "$(echo $WAIT | grep -o 'h')" ]; then
    # heures
    WAIT=$(echo $WAIT | sed 's;h;;')
    FACT=3600
else
    # secondes
    WAIT=$(echo $WAIT | sed 's;s;;')
    FACT=1
fi

WAIT=$(echo $(expr $WAIT \* $FACT))
echo $WAIT
i=0

(
    echo "#$WAIT secondes restantes"
```

```

    while [ "$i" -lt "$WAIT" ]; do
        sleep 1
        i=$(expr $i + 1)
        PERC=$(expr $i \* 100 / $WAIT )
        echo "$PERC"
        echo "#$(expr $WAIT - $i) secondes restantes"
    done
) |
zenity --progress \
    --percentage=0 \
    --title="Compte à rebours" \
    --auto-kill

aplay ~/.scripts/harp.wav
zenity --info --title="Timer" --text="C'est fini!"

exit 0

```

cclm

script de coloration des logs

[cclm](#)

```

#!/bin/sh
# colored var/log/messages in tty12
# usage: in /etc/inittab
# c12:123456:respawn:/sbin/agetty -n -l /usr/local/bin/cclm 38400 tty12
linux
file=/var/log/messages
tail -f $file | ccze -A

```

cli radio

écouter la radio depuis la console avec mplayer

[cli-radio.sh](#)

```

#!/bin/bash

#####
# cli_radio #
# based on IceFox radio-select script #

```

```
# <http://www.linuxtrack.com/t1214-radioselect-sh.htm> #
#####

# set colors
RED='\e[41m'
BLUE='\e[44m'
CYAN='\e[46m'
NC='\e[0m'

# set player
PLAYER="/usr/bin/mplayer"

# verify if $PLAYER is installed
if [ -e $PLAYER ]; then
    # stop current $PLAYER session
    if [ "$(pidof mplayer)" ]; then
        killall mplayer
    fi
    while true; do
        clear
        echo ""
        echo " ..... "
        echo " .                select a station and play                ."
        echo " .                                                                ."
        echo -e " . $CYAN[space]$NC to pause, $CYAN[q]$NC to stop,
$CYAN[Ctrl]+[c]$NC to quit ."
        echo " ..... "
        echo ""
        echo " ..... "
        echo -e " . $BLUE 1 $NC Addict Alternative $BLUE 2 $NC
Addict Lounge ."
        echo -e " . $BLUE 3 $NC Addict Rock $BLUE 4 $NC
Addict Star ."
        echo -e " . $BLUE 5 $NC Best-Hits $BLUE 6 $NC Best-
Dance ."
        echo -e " . $BLUE 7 $NC BFM $BLUE 8 $NC
Contact FM ."
        echo -e " . $BLUE 9 $NC FG DJ Radio $BLUE 10 $NC FG
Underground ."
        echo -e " . $BLUE 11 $NC FG Vintage $BLUE 12 $NC
Fréquence Jazz ."
        echo -e " . $BLUE 13 $NC Fun Radio $BLUE 14 $NC NRJ
. "
        echo -e " . $BLUE 15 $NC Radio Gerard $BLUE 16 $NC Radio
Nova ."
        echo -e " . $BLUE 17 $NC RTL $BLUE 18 $NC RTL2
. "
        echo -e " . $BLUE 19 $NC FIP $BLUE 20 $NC
custom link... ."
        echo " ..... "
        echo ""
    done
fi
```

```
echo -e " select a $BLUE number $NC from 1 to 20 then hit
$CYAN[Enter]$NC"
echo ""
read choix
case $choix in
  1)
    $PLAYER
    http://stream1.addictradio.net/addictalternative.mp3
    ;;
  2)
    $PLAYER http://stream1.addictradio.net/addictlounge.mp3
    ;;
  3)
    $PLAYER http://stream1.addictradio.net/addictrock.mp3
    ;;
  4)
    $PLAYER http://stream1.addictradio.net/addictstar.mp3
    ;;
  5)
    $PLAYER http://sv1.vestaradio.com:9500
    ;;
  6)
    $PLAYER http://sv1.vestaradio.com:7420
    ;;
  7)
    $PLAYER http://vipicecast.yacast.net/bfm
    ;;
  8)
    $PLAYER http://broadcast.infomaniak.ch/radio-contact-
high.mp3
    ;;
  9)
    $PLAYER http://fg.impek.tv/listen.pls
    ;;
  10)
    $PLAYER http://ufg.impek.tv/listen.pls
    ;;
  11)
    $PLAYER http://fgv.impek.tv/listen.pls
    ;;
  12)
    $PLAYER http://broadcast.infomaniak.ch/frequencejazz-
high.mp3
    ;;
  13)
    $PLAYER http://streaming.radio.funradio.fr:80/fun-1-44-128
    ;;
  14)
    $PLAYER http://mp3.live.tv-radio.com/nrj/all/nrj_113225.mp3
    ;;
  15)
```

```

        $PLAYER http://www.mistercouzin.net:8000/listen.pls
        ;;
16)    $PLAYER http://broadcast.infomaniak.net/radionova-high.mp3
        ;;
17)    $PLAYER http://streaming.radio.rtl.fr/rtl-1-44-96
        ;;
18)    $PLAYER http://streaming.radio.rtl2.fr:80/rtl2-1-44-96
        ;;
19)    $PLAYER http://www.tv-
radio.com/station/fip_mp3/fip_mp3-128k.m3u
        ;;
20)    echo ""
        echo "put your custom link here"
        echo ""
        read customlink
        $PLAYER $customlink
        ;;
*)    echo -e "$RED wrong choice $NC"
        echo "try again..."
        echo ""
        sleep 2
        clear
        ;;
esac
done
else
    echo " this script need mplayer"
    echo " install it or change the PLAYER"
    echo "exiting ..."
    exit 0
fi

```

gui radio

choisir sa station de radio avec zenity

[gui-radio.sh](#)

```

#!/bin/bash
#####
##                                ##
#####

```

```
#####                                IceF0x                                #####
#####
##### Script sous licence WTFPL #####
#####

#Description du programme
zenity --info \
--title="radio" \
--text="EasyRadio v1.0" \
--timeout="10"

#Choix du type d'utilisation
choixType=$(zenity --list \
--radiolist \
--title="Selectionner votre radio" \
--text="Choisissez dans la liste" \
--print-column="ALL" \
--column="Choix" \
--column="starion" \
TRUE "4U Radios - All Funky Classics" \
FALSE "Addict Alternative" \
FALSE "Choisir un fichier");

#On verifie qu'il a bien valider
case $? in
    0)

        #On lance les radio selon le choix de l'utilisateur
        if [ "$choixType" = "4U Radios - All Funky Classics" ]
        then
            mplayer http://www.4uradios.com/funkyclassics.asx
        elif [ "$choixType" = "Addict Alternative" ]
        then
            mplayer http://stream1.addictradio.net/addictalternative.mp3
        elif [ "$choixType" = "Choisir un fichier" ]
        then
            #On choisit la radio
            fileradio=$(zenity --file-selection \
--title="choisissez un fichier m3u ? ! utiliser uniquement un
format m3u");
            mplayer $fileradio
        fi

        sleep 10;
        exit;;
    1)
        echo "Annulation";
        exit;;
    -1)
        exit;;

```

```
esac
```

crypt.sh

simple script de cryptage basic

[crypt.sh](#)

```
#!/bin/bash
# crypt.sh
# source: coyotus@2012 http://www.linuxtrack.com/t660-crypt-sh.htm

echo "choisissez crypt pour créer le message ou decrypt pour décrypter le message"
read choix
if test $choix == 'crypt'
then
    read message
    echo $message | sed -e
    "y/abcdefghijklmnopqrstuvwxyz/azertyuiopqsdghjklmwxvbn/"
elif test $choix == 'decrypt'
then
    read message
    echo $message | sed -e
    "y/azertyuiopqsdghjklmwxvbn/abcdefghijklmnopqrstuvwxyz/"
else
    echo "quit"
fi
```

devart.sh

script de récupération du nombre de messages sur deviantart.com

[devart.sh](#)

```
#!/bin/bash

# devart.sh par wlourf 15/02/2011.
# script permettant de récupérer le nombre de messages sur son compte deviantArt
# lancer le script sans paramètres pour plus d'infos
# Dépend de curl

# usage devart ${texeci 600 ~/bin/devart.sh 1 messages} message(s) /
```

```
{texeci 600 ~/bin/devart.sh 0 feedbacks} feedbacks

#paramètres à renseigner (***** MASQUER LE MOT DE PASSE !!*****) :
username="*****"
password="*****"
cookie="/tmp/cookie-da.txt"
page="/tmp/da-page.html"
#fin des paramètres

#rm $cookie
#rm $page

if [[ $# < 1 ]]; then
    echo "Ce script necessite au moins un paramètre : "
    echo "  1 pour recharger la page de deviantArt"
    echo "  0 pour traiter le fichier en mémoire"
    echo
    echo "Deuxième paramètre optionnel : "
    echo "  notices, messages, feedbacks ou notes"
    echo "  si aucun choisi, le script affiche les 4 valeurs ci-dessus"
    exit 1
fi

if [[ "$1" == "1" ]]; then
    curl -k -c "$cookie" -d "username=$username&password=$password"
    https://www.deviantart.com/users/login
    curl -k -b "$cookie" -s http://my.deviantart.com/messages/ >
    "$page"
fi

function get_nb() {
    echo `cat $page | grep "View All Messages" | awk -F'i'$1'"></i>'
    '{print $2}' | awk -F'<span' '{print $1}'`
}
notices=$(get_nb 3)
messages=$(get_nb 1)
feedbacks=$(get_nb 2)
notes=$(get_nb 9)

if [[ $notices == "" ]]; then notices=0;fi
if [[ $messages == "" ]]; then messages=0;fi
if [[ $feedbacks == "" ]]; then feedbacks=0;fi
if [[ $notes == "" ]]; then notes=0;fi

case $2 in
    "notices")
        echo $notices;;
    "messages")
        echo $messages;;
    "feedbacks")
        echo $feedbacks;;

```



```
    "notes")
        echo $notes;;
    *)
        echo $notices
        echo $messages
        echo $feedbacks
        echo $notes
esac
```

dmenu-google

lancer une recherche google depuis dmenu avec votre navigateur par défaut

[dmenu_google.sh](#)

```
#!/bin/bash
entry=`echo "... and search the web" | dmenu -b -fn "snap" -nb
"#222222" -nf "#7D7D7D" -sb "#7D7D7D" -sf "#222222" -p "type a
word..."`
if test "$entry" != "X"; then
    x-www-browser http://www.google.com/search?q="$entry"
fi
```

dmenu-searx

lancer une recherche searx depuis dmenu avec surf

[dmenu_searx.sh](#)

```
#!/bin/bash
entry=`echo "... powered by searx" | dmenu -b -fn "snap" -nb "#090909"
-nf "#A0A0A0" -sb "#320000" -sf "#DEDEDE" -p "type a word then start
searching..."`
if test "$entry" != "X"; then
    surf https://searx.laquadrature.net/?q="$entry"
fi
```

dmenu-home

naviguer/ouvrir/éditer les dossiers/fichiers utilisateur depuis dmenu

[dmenu_home.sh](#)

```
#!/bin/bash
loc=`ls -A $HOME | dmenu -b -fn "snap" -nb '#222222' -nf '#7D7D7D' -sb '#7D7D7D' -sf '#222222' -p 'goto/open/edit: '` && eval "rox $loc"
```

todo-dmenu

todolist simple avec dmenu

[dmenu_todo.sh](#)

```
#!/bin/sh
## script pour gérer les différentes tâches à faire avec dmenu.
## sources:
<http://thuban.toile-libre.org/index.php/2011/01/22/un-pense-bete-avec-dmenu/>
# variables utilisées pour dmenu
SB="#7D7D7D"
SF="#222222"
NB="#222222"
NF="#7D7D7D"
FN="-*-fixed-*-*-*-*10-70-*-*-*-*-*"
FILE=~/.todo
SCRIPT=~/.bin/dmenu-todo.sh
HEIGHT=$(cat $FILE | wc -l)
PROMPT="write:add | select:del > "

ACTION="cat $FILE | dmenu -fn $FN -l '$HEIGHT' -nb '$NB' -nf '$NF' -sb '$SB' -sf '$SF' -p '$PROMPT:' "
CMD=$(eval $ACTION)
while [ -n "$CMD" ]; do
    grep -q "^$CMD" $FILE
    if [ $? = 0 ]; then
        grep -v "^$CMD" $FILE > /tmp/todo
        mv /tmp/todo $FILE
    else
        echo "$CMD" >> $FILE
        exec $SCRIPT && exit 7
    fi

    CMD=$(eval $ACTION)
done
exit 0
```

dualdetect

script basic pour le double-écran

[dualdetect.sh](#)

```
#!/bin/bash
SCREENSTATE=`xrandr | grep VGA-0 | awk '{print $2}'`
if [ "$SCREENSTATE" == "connected" ]; then
    xrandr --output VGA-0 --mode 1024x768 --pos 1024x0 --rotate normal
    --output LVDS --mode 1024x768 --pos 0x0 --rotate normal --output S-
video --off --output DVI-0 --off
else
    xrandr --output VGA-0 --off --output LVDS --mode 1024x768 --pos 0x0
    --rotate normal --output S-video --off --output DVI-0 --off
fi
exit 0
```

dzen-calendar

calendrier surgissant grâce à dzen2

[dzen_calendar.sh](#)

```
#!/bin/bash
#
# pop-up calendar for dzen
# (c) 2007, by Robert Maneaq
# mods 2012, by arpinux

TODAY=$(expr `date +%d` + 0)
MONTH=`date +%m`
YEAR=`date +%Y`

(echo '^fg(grey)' `date +%A, %d %b`; \
echo ' day:' `date +%j` - week: `date +%U`; \
echo; cal | sed -re "s/^( |[ ])*($TODAY)( |[ ]\
)/\1^fg(red)\2^fg()\3/"; \
[ $MONTH -eq 12 ] && YEAR=`expr $YEAR + 1`; cal `expr \($MONTH + 1 \)
% 12` $YEAR; \
sleep 60) | dzen2 -fn '-*-terminus-*-*-*-*12-*-*-*-*iso10646-*' -x
890 -y 740 -h 14 -w 125 -l 18 -e
'onstart=uncollapse;button1=exit;button3=exit'
```

ip publique

obtenir son ip publique

[ip.sh](#)

```
#!/bin/sh
wget -q -O - checkip.dyndns.org | sed -e 's/^[[:digit:]]\|\.//g'
```

kbd-select

sélecteur de clavier rudimentaire avec zenity

[keyboard_selector.sh](#)

```
#!/bin/bash
# simple keyboard-selector
# launch it from ~/.xinitrc
ACTION=`zenity --width=0 --height=176 --list\
    --title "SelectKbd" --text "select your keyboard"\
    --column "language"\
    "fr"\
    "gb"\
    "us"\
    "de"`

if [ -n "${ACTION}" ]; then
    case $ACTION in
        fr)
            setxkbmap fr
            ;;
        gb)
            setxkbmap gb
            ;;
        us)
            setxkbmap us
            ;;
        de)
            setxkbmap de
            ;;
    esac
fi
```

météo

sources: Nanoseb sur le [forum crunchbangfr](http://forum.crunchbangfr)

un tout petit script pour récupérer la météo en console qui utilise l'api d'[openweathermap](http://openweathermap.org).

Les dépendances sont caca-utils (pour l'affichage graphique) et libxml2-utils (pour xmllint).

la ville est à mettre en paramètre (ou alors vous pouvez aussi modifier la ligne ville="\$1")

meteo.sh

```
#!/bin/bash
#
# script météo
# sources: Nanoseb
# http://crunchbang-fr.org/forum/viewtopic.php?pid=35678#p35678
#####
fichier="$HOME/.temps"
ville=$(echo $* | sed 's/\ /%20/g')
apikey="c032679388a8e1cdf1a05b90ca396"

wget -q -O "$fichier"
"http://api.openweathermap.org/data/2.5/weather?q=$ville&mode=xml&units=metric"

temperature=$(xmllint --xpath "//current/temperature/@value" "$fichier"
| awk -F"\"" '{print $2}')
humidity=$(xmllint --xpath "//current/humidity/@value" "$fichier" | awk
-F"\"" '{print $2}')
pressure=$(xmllint --xpath "//current/pressure/@value" "$fichier" | awk
-F"\"" '{print $2}')
weather=$(xmllint --xpath "//current/weather/@value" "$fichier" | awk -
-F"\"" '{print $2}')
weathericon=$(xmllint --xpath "//current/weather/@icon" "$fichier" |
awk -F"\"" '{print $2}')
lon=$(xmllint --xpath "//current/city/coord/@lon" "$fichier" | awk -
-F"\"" '{print $2}')
lat=$(xmllint --xpath "//current/city/coord/@lat" "$fichier" | awk -
-F"\"" '{print $2}')
city=$(xmllint --xpath "//current/city/@name" "$fichier" | awk -F"\""
'{print $2}')

wget -q -O "$fichier"
"http://api.worldweatheronline.com/free/v2/tz.ashx?key=$apikey&q=$lat,$
lon&format=xml"
localtime=$(xmllint --xpath "//data/time_zone/localtime" "$fichier" |
sed 's/^<.*>\([^\<].*\)<.*>$/\1;/s/-/\ /g' | awk '{print $4 " " $3 "/"
$2 "/" $1}')

wget -q -O "$fichier"
"http://openweathermap.org/img/w/${weathericon}.png"
```

```
echo "$city          $localtime"
echo
img2txt -f utf8 -W 50 -x 7 -y 14 "$fichier" | sed '/^\s*$/d'
echo
echo "Température : $temperature°C          Humidité : $humidity%"
echo "Pression : $pressure hPa              Temps : $weather"

rm "$fichier"
```

mods

trouver les derniers fichiers modifiés

mods

```
#!/bin/bash
# find in time
# arpinix 2012 <http://arpinix.org>
if [ "$2" = "" ]; then
    param=1
else
    param=$2
fi
if [ "$1" == "-m" ]; then
    find /$PWD/ -mmin -$2
elif [ "$1" == "-d" ]; then
    find /$PWD/ -mtime -$2
else
    echo "find_in_time Usage:"
    echo "mods -m 'n' = modified in the last 'n' minutes"
    echo "mods -d 'n' = modified in the last 'n' days"
fi
```

revert

script indispensable

revert.sh

```
#!/bin/bash
# useless revert script
# sources: http://www.coyotus.com/viewtopic.php?id=1280
read message
```

```
echo $message | sed -e "y/abcdefghijklmnopqrstuvwxyz,'eqwpe+byirxlsuodbjnsin\Amx\z',/' | rev
```

le même en GUI

```
revert gui.sh
```

```
#!/bin/bash

message=$(zenity --entry --text="Veuillez entrer votre texte à retourner :" --title="Texte à l'envers")
list=$(echo $message | sed -e "y/abcdefghijklmnopqrstuvwxyz,'/eqɔpəʒɸɥıɾx̣lwsuoɒɓʝnΛmɪxλz',/' | rev)
maj=$(date +%F)

zenity --info \
--text "$list\n$maj" \
--title="Votre texte à l'envers" \
```

scratchpad

un terminal en popup

scratchpad.sh

```
#!/bin/sh

#####
# scratchpad.sh
# a simle scratchpad script by c00kielon5ter
# https://github.com/c00kiemon5ter/scripts
#####

name="scratchpad"
class="URxvt"

# print the window id of the window with the given
# instance name as the first argument '$1' and
# class name as the second argument '$2'
get_win_id() {
    xwininfo -root -children -int | awk -v n="$1" -v c="$2" 'NF < 5 {
next } $(NF - 3) == "(""n\"" && $(NF - 2) == "\"c\"")' { print $1 }'
}

# get the window id
winid="$(get win id "$name" "$class")"
```

```

# if the window was not found
if [ -z "$winid" ]
then
    # spawn it
    urxvtc -name "$name"
    # get the window id again
    winid="$(get_win_id)"
    # if the window was not found then something is really wrong. give
    up.
    [ -z "$winid" ] && exit 1
fi

# if the window is hidden show it, else hide it
if ! xwininfo -id "$winid" | awk '$1 == "Map" && $2 == "State:" { exit
($3 == "IsUnMapped") }'
then xdotool windowmap "$winid"
else xdotool windowunmap "$winid"
fi

```

screencast

enregistrer son écran depuis la console avec ffmpeg

[screencast.sh](#)

```

#!/bin/bash
# record desktop with ffmpeg

ffmpeg -r 30 -s 1024x768 -f x11grab -i :0.0 -vcodec msmpeg4v2 -qscale 2
desktop_record-`date +%Y%m%d`.avi

```

search the web

simple recherche internet lancée depuis une petite fenêtre zenity source : Photonucleon sur le [forum crunchbanglinux](#)

[ggsearch.sh](#)

```

#!/bin/bash
## google-script simple par photonucleon
## source : http://crunchbang.org/forums/viewtopic.php?id=29412
gsearch=""zenity --entry --title='Google' --text='Recherche:'`"
if [ -z $gsearch ]

```



```

    then
        exit
    else
        x-www-browser "http://www.google.fr/search?q=$gsearch"
    fi
    exit

```

session-selector

sélecteur de session sans GDM/KDM pour gestionnaire de fenêtre indépendant

[session_selector.sh](#)

```

#!/bin/bash
# simple session-selector
# launch it from ~/.xinitrc
# delete/add a section to remove/add a session
# -----
# set username
# -----
HUMAN=`w -h | tail -n1 | awk '{print $1}'`
# configure dialog window
# -----
ACTION=`zenity --width=315 --height=335 --list\
    --title "Select your Wm" --text "Welcome to Livarp_0.3.2"
$HUMAN \n pick or edit a session -->>"\
    --column "sessions" --column " what?"\
    "wmfs" "window manager from scratch v2"\
    "dwm" "the dynamic window manager reloaded"\
    "echinus" "the easy tiling"\
    "evilwm" "they sold their soul to X"\
    "pekwm" "the only one"\
    "fluxbox" "the famous one"\
    "compiz" "standalone session (testing)"\
    "" ""\
    "edit" "configure startup scripts"\
    "" ""\
    "reboot" "restart your computer"\
    "halt" "shutdown your computer"`

# session launchers
# -----
if [ -n "${ACTION}" ]; then
    case $ACTION in
        dwm)
            exec ck-launch-session $HOME/bin/start/dwm_start.sh
            ;;
        echinus)
            exec ck-launch-session $HOME/bin/start/echinus_start.sh
    esac

```

```

;;
evilwm)
    exec ck-launch-session $HOME/bin/start/evil_start.sh
;;
wmfs)
    exec ck-launch-session $HOME/bin/start/wmfs_start.sh
;;
pekwm)
    exec ck-launch-session $HOME/bin/start/pekwm_start.sh
;;
fluxbox)
    exec ck-launch-session $HOME/bin/start/fluxbox_start.sh
;;
compiz)
    exec ck-launch-session $HOME/bin/start/compiz_start.sh
;;
edit)
    geany -s $HOME/bin/start/*_start.sh
    session_selector.sh
;;
reboot)
    sudo shutdown -r now
;;
halt)
    sudo shutdown -h now
;;
esac
fi

```

shellshock-cheker

Script pour tester la présence de vulnérabilité bash CVE-2014-6271 et CVE-2014-7169

[shellshock-cheker](#)

```

#!/bin/bash
# Shellshock Vulnerability Tester Script
# Original shellshock-cheker script (C) by coyotus
# Under licence GPLv3 http://www.gnu.org/licenses/gpl.txt
# optional required "zenity"

echo ' '

```



The ASCII art logo consists of several lines of red characters forming a stylized, abstract shape. It includes horizontal and vertical lines, as well as diagonal slashes, creating a complex, geometric pattern.

```

\__ \|  Y \__ /| | | \__ \|  Y ( <_> ) \__|  <
/_/_/ \ \__|  Y \__ / \__|  < \__ /| | \
/__ >_ | / \_ >_ /_/_ /_/_ >_ | / \_ / \_ >_ | \
 \_ >_ | / \_ > \_ >_ | \_ >_ | \_ >_ | \
    \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
...

red='\e[0;31m'
green='\e[0;32m'
NC='\e[0m'

if env x='()' { :;;}; echo vulnerable' bash -c "echo -n" | grep
vulnerable > /dev/null 2>&1 ; then
    echo -e "[$red*$NC] Your system is$red vulnerable $NC"
    zenity --error --text "Your system is <span
color=\"red\">vulnerable</span>
    <a href='https://access.redhat.com/security/cve/CVE-2014-6271' >
Information</a>"
elif $(env x='()' { (a)=>\' bash -c "echo echo vuln"; [[ "$(cat echo)"
== "vuln" ]] && echo "Toujours vulnerable") | grep vulnerable ; then
    echo -e "[$red*$NC] Your system is$red vulnerable $NC"
    zenity --error --text "Your system is <span
color=\"red\">vulnerable</span>
    <a href='https://access.redhat.com/security/cve/CVE-2014-7169' >
Information</a>"
else
    echo -e "[$green*$NC] Your system is$green secure $NC"
    zenity --info --text "Your system is <span
color=\"green\">secure</span>"
fi

```

space-replace

remplacer " " par "_"

space_replace.sh

```

#!/bin/bash
#simple script to replace " " by "_"

for i in *; do
    mv "$i" "`echo "$i" | tr \ ` _`"
done

```

space-rename

alternative à space-replace par [manon](#)

[space_rename.sh](#)

```
#!/bin/bash
#simple script to replace " " by "_"
#sources: manon
<http://crunchbanglinux-fr.org/forum/viewtopic.php?id=1779>
exec rename 'y/ /_/' *
```

stats

affiche quelques info système

[stats.sh](#)

```
#!/bin/bash
# stats.sh by nlx4 <https://github.com/nlx4/bin/blob/master/stats.sh>

clear
declare -rx BC="/usr/bin/bc"

if test -f $HOME/.gtkrc-2.0;then
    GTK=$HOME/.gtkrc-2.0
else
    GTK=$HOME/.gtkrc-3.0
fi

if test ! -x $BC; then
    echo -e "\e[0;5;31m$BC is not available, please install before continuing."
    exit 192
fi

host=`cat /proc/sys/kernel/hostname`
mem_total=`free -m | grep "Mem" | awk '{print $2}'`
mem_used=`free -m | grep "buffers/" | awk '{print $3}'`
hd_used=`df | grep "sda" | awk '{print $3}'`
hd_total=`df | grep "sda" | awk '{print $2}'`
type=`df -T | grep "sda" | awk '{print $2}'`
idle=`cat /proc/uptime | awk '{print $2}'`
kernel=`cat /proc/version | awk '{print $3}'`
cpu=`cat /proc/cpuinfo | grep "model name" | awk '{print $4,$5,$6,$7}'`
```

```

theme=`cat $GTK | grep "gtk-theme-name" | cut -d= -f2 | tr -d '`
icon=`cat $GTK | grep "gtk-icon-theme-name" | cut -d= -f2 | tr -d '`
font=`cat $GTK | grep "gtk-font-name" | cut -d= -f2 | tr -d '`

function uptime() {
    uptime=$(</proc/uptime)
    uptime=${uptime%.*}

    seconds=$(( uptime%60 ))
    minutes=$(( uptime/60%60 ))
    hours=$(( uptime/60/60%24 ))
    days=$(( uptime/60/60/24 ))
    echo -e $days"-days", $hours"-hrs" $minutes"-mins" $seconds"-s"
}

echo -e "\e[1;33mUptime\e[0m:    $(uptime)"
echo -e "\e[1;33mKernel\e[0m:    $kernel"
echo -e "\e[1;33mHostname\e[0m: $host"
echo -e "\e[1;33mCPU\e[0m:        $cpu"
echo -e "\e[1;33mRAM\e[0m:         $mem_used Mb / $mem_total Mb
(\e[0;31m$((($mem_used*100/$mem_total))\e[0m%)"
echo -e "\e[1;33mROOT\e[0m:         $(echo "scale=3; $hd_used/1000000" |
bc) Gb / $(echo "scale=3; $hd_total/1000000" | bc) Gb
(\e[0;31m$((($hd_used*100/$hd_total))\e[0m%) ($type)"
echo -e "\e[1;33mTheme\e[0m:     $theme"
echo -e "\e[1;33mIcons\e[0m:      $icon"
echo -e "\e[1;33mFont\e[0m:       $font"
echo -e "\n"
echo -e "Colors:\n\e[0;30m##### \e[0;31m##### \e[0;32m#####
\e[0;33m##### \e[0;34m##### \e[0;35m##### \e[0;36m#####
\e[0;37m#####\e[0m"
echo -e "\e[1;30m##### \e[1;31m##### \e[1;32m##### \e[1;33m#####
\e[1;34m##### \e[1;35m##### \e[1;36m##### \e[1;37m#####\e[0m\n"
exit 0

```

server mods

recevoir par mail les derniers fichiers modifiés sur un serveur

[watch_server.sh](#)

```

#!/bin/bash
#
#
#
#

echo -e "

```



```

    "
for b in {0..8}; do
    ((b>0)) && bg=$((b+39))

    echo -en "\033[0m ${BGNames[b]} | "

    for f in {0..7}; do
        echo -en "\033[${bg}m\033[$((f+30))m ${FGNames[f]} "
    done

    echo -en "\033[0m | "
    echo -en "\033[0m\n\033[0m      | "

    for f in {0..7}; do
        echo -en "\033[${bg}m\033[1;${(f+30)}m ${FGNames[f]} "
    done

    echo -en "\033[0m | "
    echo -e "\033[0m"

    ((b<8)) &&
    echo "
|
|
done
echo "
|
|
"

```

termaliencolors

afficher les couleurs du terminal avec des aliens

[termaliens_color.sh](#)

```

#!/bin/bash
#
# ANSI color scheme script featuring Space Invaders
#
# Original: http://crunchbanglinux.org/forums/post/126921/#p126921
# Modified by lolilolicon
#

f=3 b=4
for j in f b; do
    for i in {0..7}; do
        printf -v $j$i %b "\e[${!j}]${i}m"
    done
done

```

```

done
bld=$'\e[1m'
rst=$'\e[0m'

cat << EOF

    $f1  $f2  $f3  $f4  $f5
    $f6  $rst
    $f1  $f2  $f3  $f4
    $f5  $f6  $rst
    $f1  $f2  $f3  $f4
    $f5  $f6  $rst
    $f1  $f2  $f3  $f4  $f5
    $f6  $rst

    $bld$f1  $f2  $f3  $f4
    $f5  $f6  $rst
    $bld$f1  $f2  $f3  $f4
    $f5  $f6  $rst
    $bld$f1  $f2  $f3  $f4
    $f5  $f6  $rst
    $bld$f1  $f2  $f3  $f4
    $f5  $f6  $rst

    $f7  $rst

    $f7  $rst

    $f7  $rst
    $f7  $rst
    $f7  $rst

EOF

```

watch-inside

visionneur basic avec feh

[watch-inside.sh](#)

```

#!/bin/bash
# image browser with feh
# sources: https://bbs.archlinux.org/viewtopic.php?pid=884635#p884635
# Change the input field separator to avoid problems with spaces.
IFS='
'

# Get the target file, its name, directory, and the other files in the

```



```

directory.
_fpath="${1%/\/}"
_fname="$(basename "$_fpath")"
_dpath="$(dirname "$_fpath")"
_flist=( $(ls "$_dpath") )

# Determine the index of the target file in the file list array.
_i=0
for _file in "${_flist[@]}; do
    [ "$_file" == "$_fname" ] && break
    _i=$(( _i + 1 ))
done

# Remove the target file from the arguments, leaving any eventual
arguments to
# feh.
shift
# Change to the directory and launch feh with the target file as the
first
# argument, and the rest of the files in their normal order.
cd "$_dpath"
feh -Z -g 600x600 -B black -q "$@" "${_flist[@]:$_i}"
"${_flist[@]:0:$_i}"

```

xcomp-select

sélecteur d'effet pour xcompmgr

[xcomp-select.sh](#)

```

#!/bin/bash
# xcompmgr level selector
# arpinux 2012 <http://arpinux.org>
if [ "$1" == "-s" ]; then
    killall xcompmgr
elif [ "$1" == "-l" ]; then
    killall xcompmgr
    sleep 1s
    xcompmgr -fF &
elif [ "$1" == "-m" ]; then
    killall xcompmgr
    sleep 1s
    xcompmgr -CcF -I20 -O10 -D1 -t-5 -l-5 -r4.2 -o.82 &
elif [ "$1" == "-f" ]; then
    killall xcompmgr
    sleep 1s
    xcompmgr -CcF -I-.015 -O-.03 -D6 -t-1 -l-3 -r4.2 -o.5 &
else

```

```
echo "xcompmgr_livarp Usage:"  
echo "-s = stop"  
echo "-l = light"  
echo "-m = medium"  
echo "-f = full"  
  
fi
```

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/utilisateurs:arpinux:tutos:scripts-f-utiles>



Last update: **03/07/2016 00:25**