
- CUSTOMIZE YOUR SHELL MOTD -

When users login to their shell account on a Linux they will be first greeted with uname, followed by the MOTD (message of the day), and then mail if they have any and then the last login information.

This tutorial will show you how to remove and or change those and add an informative system information and some fun stuff like weather and a daily random quote or fortune. Step by step; you can do it! :)

[DEFAULT MOTD](#)

[REMOVE DEFAULT MOTD](#)

[REMOVE OR MODIFY UNAME](#)

[LAST LOGIN](#)

[CUSTOMIZE YOUR MOTD](#)

[1. TCLSH](#)

[2. HDDTEMP](#)

[3. LM-SENSORS](#)

[4. WEATHER](#)

[5. EASY FAST ASCII ART](#)

[EXPLAIN A BIT ABOUT THE MOTD SCRIPT CODE](#)

[CREATE THE MOTD SCRIPT](#)

[TEXT FORMAT/COLOR](#)

[DEGREE SYMBOL ° PROBLEM](#)

[RANDOM FORTUNE OR A QUOTE ON EACH LOGIN](#)

[COWSAY THE FORTUNE](#)

OTHER MOTD IDEAS

_____.._____

For Example Debian's default MOTD is this:

Linux yourserver.com 1.2.34-5-678 #1 Wed Nov 4 20:19:07 UTC 2009 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

You have mail.

Last login: Sat Apr 24 07:30:23 2010 from 123.456.78.90

mewbies@whatever:~\$

|-----..-----|

This is now my MOTD:

You have mail.

 - -
 | | ()
 | | - -
 - - - - -
| ' _ ` _ \ / _ \ \ / \ / / ' _ \ | / | / _ \ / _ |

```
| | | | | | _/\ v v /| |_) | | _/\_ \
|_| |_| |_| \__| \_/\_/ |_._/|_| \__||__/
```

Last Login.....: Mon Jun 14 02:59:55 from 123.456.78.90

Uptime.....: 50days 15hours 51minutes 52seconds

Load.....: 0.02 (1minute) 0.01 (5minutes) 0.00 (15minutes)

Memory MB.....: 488 Used: 457 Free: 30 Free Cached: 295 Swap In Use:
160

Temperature...: Core0: +91.4°F M/B: +98.6°F CPU: +89.6°F Disk: 98°F

Disk Usage.....: You're using 2MB in /home/mewbies

SSH Logins.....: There are currently 3 users logged in.

Processes.....: You're running 4 which makes a total of 84 running

Weather.....: 58°F, Cloudy

.....-STATEMENT-

.....
This is a private system that you are not to give out access to anyone
without permission from an admin!

/ You will be Told about it Tomorrow. Go \
\ Home and Prepare Thyself. /

 \ ,__/
 \ (oo)____
 ())\
 ||--|| *

mewbies@whatever:~\$



Tip: Create a new user, or use an existing one, that isn't in the *admin group, for testing purposes - this way you don't need to exit root, logout, login each time to view the changes as we go.

*The reason to test using a user that isn't in the admin group is that when a user logs in, the cmds that are on the MOTD script are actually being issued by the user, so if you have a cmd that isn't allowed for a user to issue MOTD will show only errors. (All the statements above on my MOTD, at least for my setup, are available to all users.)

REMOVE DEFAULT MOTD:

To remove the lines or change the MOTD section which is "The programs included ... by applicable law." you need to (commands are in bold):

```
su
pico /etc/motd
```

Remove everything, as I have done (or alter it to your liking).

Log back in to view the changes.

This change though is only temporary - next time the box reboots the MOTD will be restored back to its previous state by the bootmisc.sh, which is here /etc/init.d/bootmisc.sh, as it re-writes back to /var/run/motd the contents of /etc/motd.tail - etc/motd is actually a symbolic link /var/run/motd.

(bootmisc.sh contains:

```
# Update motd
uname -snrvm > /var/run/motd
[ -f /etc/motd.tail ] && cat /etc/motd.tail >> /var/run/motd )
```

So to make this change permanent, you need to remove everything (or to your liking) also from:

pico /etc/motd.tail

_____.._____

REMOVE OR MODIFY UNAME:

.....

If you want to remove the 'uname' (print system information) - the first line printed out when you login, similar to:

Linux yourserver.com 1.2.34-5-678 #1 Wed Nov 4 20:19:07 UTC 2009 i686

Do this:

su

pico /etc/init.d/bootmisc.sh

Change this line, as I have done:

```
uname -snrvvm > /var/run/motd
```

To:

```
# uname -snrvvm > /var/run/motd
```

You will need to reboot for the uname changes to take place; this is the only modification we'll make that requires a reboot to view the changes:

reboot

Next time you login you will only see:

You have new mail.

Last login: Sat Apr 24 07:30:23 2010 from 12.34.56.78

Or JFYI (not needed for our MOTD) if you want the uname to contain different information, view options here:

man uname

q

You can view the options by running the cmd, for example:

uname -o

Output would be similar to:

GNU/Linux

Then edit the options:

su

pico /etc/init.d/bootmisc.sh

Change this line:

```
uname -snrvn > /var/run/motd
```

To, for example:

```
uname -o > /var/run/motd
```

_____.._____

LAST LOGIN:

.....

I do not recommend removing the 'Last login' statement because if someone did gain access to your account you'll see an IP that isn't yours (if they didn't cover their tracks that is). So what I have done is removed it from the standard MOTD and added it to our MOTD script below so that visually it looks better with the style I have. If you do want to remove it:

pico /etc/ssh/sshd_config

Find this: PrintLastLog yes

Change to: PrintLastLog no

Restart your ssh server for the changes to take affect:

/etc/init.d/ssh restart

NOTES:

.....

1. You don't need to put everything I have above on my MOTD - I just added a variety of information to give some ideas.

In our MOTD script if you want to remove something it will be in two places; the '# MOTD script' section (that issues the cmds) and the

'# * Print Results' section that, well as it says prints the results of

the cmds :).

_____.._____

CUSTOMIZE YOUR MOTD:

.....

PREREQUISITES:

1. **TCLSH:** View if you have tclsh installed:

dpkg -l | grep tcl

Reply if you do have it installed will be similar to:

tcl8.5 8.5.3-2 Tcl (the Tool Command Language) v8.5 - run-t

tcl8.5-dev 8.5.3-2 Tcl (the Tool Command Language) v8.5 - devel

If not:

su

aptitude install tcl

_____.._____

2. **HDDTEMP:** If you want temperature readings for your hard drive, using

'hddtemp', your disk must have SMART technology (most do for the past few years I've read).

(If you want to know more about [hddtemp](#): **man hddtemp**)

Do this:

apt-get install hddtemp

Select <Yes> to all the questions asked by using your arrow or tab key

then hitting Enter. You can easily modify these settings later if needed.

(You'll be asked 3 questions: to run as a daemon, to listen on local IP, and what port). After it is finished it will reply similar to:

```
Starting disk temperature monitoring daemon: hddtemp: /dev/hda.
```

It will run as a daemon (run levels 2-5).

To modify any of these settings:

```
dpkg-reconfigure hddtemp
```

or

```
pico /etc/default/hddtemp
```

Now test hddtemp cmd:

```
hddtemp -v
```

If you receive 'hddtemp: command not found' run this:

```
dpkg-reconfigure hddtemp
```

answer Yes to all

Then try it again:

```
hddtemp -v
```

Now you need to work out the correct cmd for your drive and the output

that you would like. Once you have it make a note of the cmd for our MOTD

script. I have the output to be only, for example, 35°C.

```
hddtemp /dev/hda
```

Or yours might be: **hddtemp /dev/sda**

Output will be similar to:

```
/dev/hda: WDC WD400BB-22JHC0: 35°C
```

Replace the 'a' in hda or sda with the drive letter you want hddtemp to

report on.

Some examples of hard drive names:

```
/dev/hda  master device on primary IDE channel
/dev/hdb  slave device on primary IDE channel
/dev/hdc  master device on secondary IDE channel
/dev/hdd  slave device on secondary IDE channel
/dev/sda  first SCSI hard drive
/dev/sdb  second SCSI hard drive
```

So to shorten the reply to be only '35°C' count the characters from the beginning of the reply until where you want it to reply. Using my example

```
output: /dev/hda: WDC WD400BB-22JHC0: 35°C
```

the cmd would be, to reply only the 31st - 35th characters:

```
hddtemp /dev/hda | cut -c 31-35
```

Try the cmd yourself to understand what I mean, replacing disk name, hda, with yours.

Or to display temperature in Fahrenheit:

```
hddtemp /dev/hda -uf | cut -c 31-35
```

Or to shorten the reply to temperature only e.g. 35:

```
hddtemp -n /dev/hda
```

_____.._____

3. LM-SENSORS: If you want temperature readings for your other devices, amongst other information, I used lm-sensors.

(If you want to know more about [lm-sensors](#): **man lm-sensors** press q to quit)

```
apt-get install lm-sensors
```

After install is complete:

```
sensors-detect
```

Hit enter key when prompted YES/No questions to select the default answer for all. (You can easily modify these by running sensors-detect again if needed later.)

If sensors-detect hangs on a sensor; make a note which one, then press Ctrl+c to quit sensors-detect.

Then run sensors-detect again and answer No to the one that it hung on.

At the end you'll have been prompted to add lines to your /etc/modules.

This is the reply I received:

To load everything that is needed, add this to /etc/modules:

```
#----cut here----
```

```
# Chip drivers
```

```
w83627hf
```

```
k8temp
```

```
#----cut here----
```

Now you need to add the output that it replied with, as shown above:

```
pico /etc/modules
```

Mine has this:

```
loop
```

Changed it to, using example above, (add your own output):

```
loop
```

```
# Chip drivers
```

```
w83627hf
```

```
k8temp
```

```
Now start it:
```

```
/etc/init.d/module-init-tools
```

```
Output: Loading kernel modules...done.
```

```
JFYI:
```

```
/etc/init.d/module-init-tools start
```

```
/etc/init.d/module-init-tools stop
```

```
/etc/init.d/module-init-tools restart
```

```
/etc/init.d/module-init-tools force-reload
```

```
By the way it has a run level 'S' - it will run at start up on its own.
```

```
Now to load the modules that you have added above to /etc/modules,  
changing the module name below to your own, for the example above, I  
would run:
```

```
modprobe w83627hf
```

```
modprobe k8temp
```

```
And then to update the modules (previously known as update-modules):
```

```
depmod -a
```

```
After the update is finished, test lm-sensors by running:
```

```
sensors
```

```
The output should be similar to:
```

```
k8temp-pci-00c3
```

```
Adapter: PCI adapter
```

```
Core0 Temp: +29.0°C
```

```
w83627thf-isa-0290
```

```
Adapter: ISA adapter
```

```
VCore:      +1.43 V (min = +0.70 V, max = +1.87 V)
+12V:      +11.98 V (min = +1.95 V, max = +6.02 V)  ALARM
+3.3V:      +3.36 V (min = +0.00 V, max = +1.58 V)  ALARM
+5V:       +4.99 V (min = +1.81 V, max = +3.84 V)  ALARM
-12V:      -11.46 V (min = -14.91 V, max = -4.38 V)
V5SB:      +5.05 V (min = +2.85 V, max = +0.43 V)  ALARM
VBat:      +3.49 V (min = +3.62 V, max = +0.02 V)  ALARM
fan1:      2721 RPM (min = 8035 RPM, div = 2)  ALARM
CPU Fan:   3054 RPM (min = -1 RPM, div = 2)  ALARM
fan3:      0 RPM (min = 168750 RPM, div = 2)  ALARM
M/B Temp:  +36.0°C (high = +5.0°C, hyst = +0.0°C)  ALARM  sensor =
CPU Temp:  +30.5°C (high = +60.0°C, hyst = +55.0°C)  sensor =
temp3:     +65.5°C (high = +60.0°C, hyst = +55.0°C)  ALARM  sensor =
cpu0_vid:  +0.000 V
beep_enable:enabled
```

The cmd I use on our MOTD script to report only the temperature for the Core0, M/B (mother board) and the CPU is:

```
sensors -f | grep Temp
```

To alter that output so that it is like my MOTD above, I've done that in the MOTD script itself as you'll see soon.

If you want the output to be in Celsius:

```
sensors| grep Temp
```

4. WEATHER:

If you would like the Weather report on your MOTD like I have above, you'll need to install curl. To view if you have it installed already:

```
dpkg -l | grep curl
```

If you do output will be similar to:

```
curl 7.18.2-8lenny4 Get a file from an HTTP, HTTPS or FTP server
libcurl3 7.18.2-8lenny4 Multi-protocol file transfer library (OpenSS
```

If you don't have either:

```
aptitude install curl libcurl3
```

Next you need to find the correct 'location code' for the area you want the weather to report on. Go [HERE](#). **UPDATE 01 JUL '13:**

accuweather.com has killed that URL, made it very difficult to find location codes and doesn't provide rss URLs for locations. Not to worry - I tested this method with over twenty cities and it works.

This is how to find accuweather.com location codes:

United States use your zip code.

Non US (and United States to be sure they have the weather for X zip code)

go [HERE](#), fill in the location etc, finish the steps; you do not need to sign up. At the end you'll be given an HTML code for your location. View the last part of the HTML code for the location code. For example Mexico City, Mexico results:

```
[snip] ?partner=accuweather&zipcode=NAM|MX|MX009|MEXICO CITY| ' [snip]
```

The location code would be: NAM|MX|MX009|MEXICO CITY

For New York, NY for example it would be: 10001

To turn the location codes into a rss URL, city names with more than one word replace spaces with a dash or %20 or remove the space:

[http://rss.accuweather.com/rss/liveweather_rss.asp?metric=0&locCode="+code](http://rss.accuweather.com/rss/liveweather_rss.asp?metric=0&locCode=)

For example:

http://rss.accuweather.com/rss/liveweather_rss.asp?metric=0&locCode=NAM|MX|MX009|MEXICO-CITY

http://rss.accuweather.com/rss/liveweather_rss.asp?metric=0&locCode=NAM|MX|MX009|MEXICO%20CITY

http://rss.accuweather.com/rss/liveweather_rss.asp?metric=0&locCode=NAM|MX|MX009|MEXICOCITY

Test that your rss feed URL works, 0 for F and 1 for C (Celsius):

http://rss.accuweather.com/rss/liveweather_rss.asp?metric=0&locCode=10001

Now to create a script to gather the weather data in a format that you want, with thanks to the codes [HERE](#) by fflarex & fukawi2, do this:

Find the location of your sh:

which sh

Output will be the location of your sh, for example my output is:

/bin/sh

Or your output might be:

/usr/bin/sh

Then paste in the script below changing three things:

1. '/bin/sh' change this to the location of your sh if it's not /bin/sh
2. Output in Celsius change 0 to 1 here: METRIC=0 # 0 for F, 1 for C
3. Fill in your location code here: LOCCOD=""

For example: LOCCOD="NAM|MX|MX009|MEXICO-CITY"

```

■ pico /usr/share/weather.sh
■
#!/bin/sh

METRIC=0 # 0 for F, 1 for C
# Fill in form to find your weather code here:
# http://netweather.accuweather.com/signup-page2.asp
# If code has a space remove it or replace it with %20 or a dash; -
LOCCOD="" #Example: NAM|MX|MX009|MEXICO-CITY

if [ -z $1 ] && [ -x $LOCCOD ] ; then
    echo
    echo "USAGE: $0 [locationcode]"
    echo
    exit 0;
elif [ ! -z $1 ] ; then
    LOCCOD=$1
fi

curl -s
http://rss.accuweather.com/rss/liveweather_rss.asp?metric\=${METRIC}\&locCod
e\=$LOCCOD \
| sed -n '/Currently:/ s/.*: \(.*\): \([0-9]*\)\([CF]\).*\/\2°\3, \1/p'

```

■ Set perms on your weather.sh:

```
■ chmod 744 weather.sh
```

■ Then test it:

```
■ /usr/share/./weather.sh
```

■ Output:

```
■ 56°F, Thunderstorm
```

_____.._____

■ 5. EASY FAST ASCII ART:

■ To make an ASCII art design like I have done above for 'mewbies' either:

■ Visit ruletheweb.co.uk, type in what you want, copy, paste, add a margin

■ on the left. I used the font doom. Looks better if you don't surpass 80

■ 80 characters per line.

■ Or:

■ Install [FIGlet](http://figlet.com) and make just like on ruletheweb but with more choices and

in your own terminal.

_____.._____

EXPLAIN A BIT ABOUT THE MOTD SCRIPT CODE:

If you know the how to use the code in the script below, skip this. For those that it is foreign I'll explain a bit to help you adjust it.

For example the output of the `lm-sensors` cmd '`sensors -f | grep Temp`' we did earlier needs to be modified to suit what we actually want on our MOTD. Below in the script, it has the output to MOTD based on the results of my output from the cmd. You might need to change this depending on what your own output from that cmd is.

My output is:

Core0 Temp: +89.6°F

M/B Temp: +100.4°F (high = +41.0F, hyst = +32.0F) ALARM sensor = thermistor

CPU Temp: +90.5°F (high = +140.0F, hyst = +131.0F) sensor = thermistor

I only want it to output: +89.6°F & +100.4°F & +90.5°F

This is done by using '`lindex`' and counting the place of a complete word that you want on the output, not including white spaces, starting with 0 (zero). So for example +89.6°F is the 3rd word, since we start counting at 0, it would be in place 2, then +100.4°F would be in place 5, and so on.

The code in the script below to output only the 2, 3, & 18 place word is:

```
set temperature      [exec -- sensors -f | grep Temp]
set tem(0)           [lindex $temperature 2]
set tem(m)           [lindex $temperature 5]
set tem(c)           [lindex $temperature 18]
```

'tem(0)' can be any name() you like as long as you use the same name() in the output section.

[lindex \$temperature 2] is the code that will output the place 2 word (3rd word in the output).

Then in our output section will use tem(0), tem(m) etc with descriptions for the output.

If that seems more confusing, don't worry about it, just use the script then I you'll have a better understanding.

_____.._____

CREATE THE MOTD SCRIPT:

.....

Paste in the script below changing the code to suit your cmds- the notes you made earlier, such as:

1. Your correct disk name for hddtemp along with the count of the characters to output out on this line:

```
set hddtemp [lindex [exec -- /usr/bin/hddtemp /dev/hda -uf | cut -c "31-35"] 0]
```

2. The location of 'env':

which env

My output is: /usr/bin/env

If yours is different change in the first line of the script to your path

OK so lets do this- 1. Create a new file 'motd.tcl':

su

pico /etc/motd.tcl

Copy/paste in:

```
#!/usr/bin/env tclsh
# MOTD script original? / mod mewbies.com

# * Variables
set var(user) $env(USER)
set var(path) $env(PWD)
set var(home) $env(HOME)

# * Check if we're somewhere in /home
#if {[string match -nocase "/home*" $var(path)]} {
if {[string match -nocase "/home*" $var(path)] && ![string match -nocase
"/usr/home*" $var(path)] } {
    return 0
}

# * Calculate last login
set lastlog [exec -- lastlog -u $var(user)]
set ll(1) [lindex $lastlog 7]
set ll(2) [lindex $lastlog 8]
set ll(3) [lindex $lastlog 9]
set ll(4) [lindex $lastlog 10]
set ll(5) [lindex $lastlog 6]

# * Calculate current system uptime
set uptime [exec -- /usr/bin/cut -d. -f1 /proc/uptime]
set up(days) [expr {$uptime/60/60/24}]
set up(hours) [expr {$uptime/60/60%24}]
set up(mins) [expr {$uptime/60%60}]
set up(secs) [expr {$uptime%60}]

# * Calculate usage of home directory
set usage [lindex [exec -- /usr/bin/du -ms $var(home)] 0]

# * Calculate SSH logins:
set logins [exec -- w -s]
set log(c) [lindex $logins 5]

# * Calculate processes
set psu [lindex [exec -- ps U $var(user) h | wc -l] 0]
set psa [lindex [exec -- ps -A h | wc -l] 0]

# * Calculate current system load
set loadavg [exec -- /bin/cat /proc/loadavg]
set sysload(1) [lindex $loadavg 0]
set sysload(5) [lindex $loadavg 1]
set sysload(15) [lindex $loadavg 2]

# * Calculate Memory
set memory [exec -- free -m]
set mem(t) [lindex $memory 7]
set mem(u) [lindex $memory 8]
set mem(f) [lindex $memory 9]
set mem(c) [lindex $memory 16]
```



```
close $fp
puts ""
}
```

2. Set permissions on the file:

```
chmod 755 /etc/motd.tcl
```

3. Then add your new motd.tcl to the user's profile:

```
pico /etc/profile
```

Add this line at the end of profile:

```
/etc/motd.tcl
```

4. Logout and back in to view the changes, or exit root then:

```
/etc/./motd.tcl
```

_____.._____

TEXT FORMAT/COLOR NOTES:

The mewbies ASCII and the 'Statement' is in light green as stated with this line for the ASCII:

```
puts "\033\[01;32m$head\033\[0m" :
```

\033\[starts code, 01 is light, 32 is green, m is format, \$head is the mewbie's ASCII, \033\[0m turns off the text formatting.

=====.

CONSOLE CODES CHART			
COLOR	TEXT BACKGROUND	COLOR	TEXT

Black	30	40	Dark Gray	1;30
Red	31	41	Light Red	1;31
Green	32	42	Light Green	1;32
Yellow	33	43	Light Yellow	1;33
Blue	34	44	Light Blue	1;34
Magenta	35	45	Light Magenta	1;35
Cyan	36	46	Light Cyan	1;36
Light Gray	37	47	White	1;37
FORMAT		FORMAT		
reset	0		underscore on,	
bold	1	default foreground color	<u>38</u>	
half-bright	2		underscore off,	
<u>underline</u>	4	default foreground color	39	
blink	5	default background color	49	
inverse	7			
conceal	8			
normal	22		man console_codes	
underline off	24			
blink off	25			
inverse off	27		mewbies.com	

"=====

CONSOLE COLOR CODES

		40m	41m	42m	43m	44m	45m	46m	47m
m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
1m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
30m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
1;30m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
31m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
1;31m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
32m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
1;32m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
33m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
1;33m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
34m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
1;34m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
35m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
1;35m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
36m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
1;36m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
37m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew
1;37m	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew	Mew

[HERE](#) are the Console Codes against black.

If you would like the script to output the above color charts:

```
wget http://mewbies.com/geek_fun_files/color_scripts/mewbies_colors.zip
unzip mewbies_colors.zip && chmod 755 mewbies_colors.sh
./mewbies_colors.sh
```

View [HERE](#) for more color scripts to demonstrate your console's color capabilities.

For a visual understanding enter this in your shell:

```
echo -e "\033[08mI'm concealed\033[0m"
echo -e "\033[32mI'm green\033[0m"
echo -e "\033[01mI'm brighter-bold\033[0m"
echo -e "\033[01;32mI'm light green\033[0m"
echo -e "\033[01;32;45mI'm light green on a magenta background\033[0m"
echo -e "\033[01;04;32;45mI'm light underlined green on a magenta background\033[0m"
echo -e "\033[01;04;32;45;5mI'm light underlined green on a magenta background, blinking\033[0m"
echo -e "\033[07;05;04;01;32;45mI'm inversed; underlined blinking magenta on a light green background\033[0m"
```

At this point you might want to enter:

```
clear
```

Example Background: 1;7;40

Since the attribute 1 is used on the text to lighten (bolder) color you need to use the inverse with it if you want to use it on the background color. Some examples of using 1 - lightener with 7 - inverse:

```
echo -e "\033[30;47mI'm black on a light gray background\033[0m"
```

But I want to be black on dark gray background.

So we need to use 1 and 7 so that the background becomes dark gray by stating the opposite - text black 30 and background black 40:

```
echo -e "\033[1;7;30;40mI'm black on a dark gray background\033[0m"
```

Where as to be black on white background we state text light gray (1 will change that to white and 7 will inverse it to be the background not the text) on a black background:

```
echo -e "\033[1;7;37;40mI'm black on a white background\033[0m"
```

Once you get your head around that, try this:

```
echo -e "\033[1;7;30;37mI'm black on a white background too\033[0m"
```

Change the order of the 30 & 37:

```
echo -e "\033[1;7;37;30mI'm black on a dark gray background\033[0m"
```

If the formatted text isn't showing properly it is most likely because the SSH client you are using doesn't have colors enabled. To enable them for example in -

SecureCRT: Options/Session/Terminal/Emulation/ check the box 'ANSI Color'.

Terminal has VT100 by default which will work fine.

PuTTY: Will have colors on by default (Window/Colours) but not blinking text. To turn it on Terminal/check the box 'Enable blinking text'.

MOTD COLOR HELPER TOOLS:

If you are also a Windows user I've found a great freeware, tiny, portable program 'MOTD Maker' [HERE](#) to colorize and format text easily.

_____.._____

DEGREE SYMBOL ° PROBLEM:

.....

If your degree symbol ° doesn't show correctly, such as 'Â°' read [HERE](#) how to fix this.

_____.._____

RANDOM FORTUNE OR A QUOTE ON EACH LOGIN:

.....

You could have a random fortune or a quote printed out to users each time a user logs in. Here is one from fortune:

Good day for overcoming obstacles Try a steeplechase.

su

apt-get install fortune

Then to add fortune's path to the bottom of profile:

pico /etc/profile

Add this line:

/usr/games/fortune

exit

Now each time a user logs in it will execute fortune, giving them a random quote.

To view fortune's options:

man fortune

To leave the man window:

q

To customize Fortune more visit my Fortune tutorial [HERE](#) under GEEK FUN.

COWSAY THE FORTUNE:

.....

You could be sillier and have one of cowsay's creatures state the quote for them:

su

apt-get install cowsay

Instead of adding this line to the bottom of the profile file:

/usr/games/fortune

Add this line for example:

/usr/games/fortune | cowsay -f small

The output of that would be:

```
/ Good day for overcoming obstacles. Try \  
\ a steeplechase. /
```

```
 \  '___'  
 \  (oo)____  
   (  )    )\  
     ||--|| *
```

If you received the error:

cowsay: Could not find small cowfile!

Either use a different cow (creature) or fix it by going [HERE](#).

To find out more including variables for cow's tongue, eyes, position,
etc.:

man cowsay

You can view the output before adding the line by running it:

fortune | cowthink -f www

fortune | cowsay -f bud-frogs

fortune | cowsay -f skeleton

etc...

To customize cowsay more visit my Cowsay tutorial [HERE](#) under GEEK FUN.

_____.._____

OTHER MOTD IDEAS:

.....

1. Just add these lines to your motd.tcl if you for example you're the

only one logging in- as these cmds aren't available to regular users- MOTD will only produce errors if a regular user logs in with these on the motd.tcl.

To have on your MOTD report your week's total times you have logged in:

```
# * Calculate SSH logins:
set userwktotal [lindex [exec -- grep opened /var/log/auth.log | awk
/$var(user)/ | wc -l] 0]
puts " SSH Logins....: There are currently $log(c) users. You have logged
in a total of ${userwktotal} times this week"
```

To report the week's total SSH login failed attempts:

```
set failures [lindex [exec -- grep sshd /var/log/auth.log | awk
/failure/ | wc -l] 0]
puts " SSH Failed....: There have been ${failures} failed attempts this
week to possibly break-in"
```

Depending on how often your logs are rotated will be if the report is weekly or other. To view when your logs are rotated:

```
cat /etc/logrotate.conf
```

Mine has:

```
# rotate log files weekly
weekly
```

One way around these type of admin only cmds for all users to have on their MOTD is to create a bash file with those cmds with the output placed in a text file, with the correct permissions set, in a users shared directory, then cron the bash fail to run every x minutes and then add to the motd.tcl code to cat the text file -not execute the actual cmd.

For example:

```
grep sshd /var/log/auth.log | awk /failure/ | wc -l > failure_count.txt
cat failure_count.txt
```

the '>' will over write the file each time

if it was '>>' it would append the output to the file.

I don't really fancy this method as it just means another process running every x minutes and that it's not really up to the minute information if set for longer periods.

2. Just like we added the fortune and cowsay to /etc/profile you could add for example your weather script to /etc/profile like this:

```
/usr/share/./weather.sh
```

3. If you wanted each user to have their own weather (their own location code) reported on their MOTD and not the location code you have on the main one you could do it like this:

In each users home directory there must be a weather.sh with their own desired location code, chmod 744. Then on the motd.tcl, this line:

```
set weather      [exec -- /usr/share/./weather.sh]
```

Change to:

```
set weather      [exec -- /home/$var(user)/./weather.sh]
```

4. Use [Linux Logo](#) to display system information and your OS's logo.

5. If you want your MOTD to stay on top of your terminal window:

```
clear; /bin/echo -e '\033[30;70r' ; /etc/./motd.tcl
```

Turn it off: `/bin/echo -e '\033[0;0r'`

Adjust '30' for how many rows (lines) the MOTD is, '70' is work space.

If you know of any other fun or useful variables to add to MOTD please

post then at mewbies [forum](#) :)

//-----

If you find mistakes, have suggestions, and or questions please post at
mewbies forum [HERE](#) - thank you.

Last update on 01 Jul '13

- mewbies.com -