ANTIX LINUX SETUP GUIDE

-- First Download Antix-core iso and burn it on a pendrive using dd command. Do not use isohybrid.

-- If you want to install it alongside windows, it is better to install it on a separate EFI partition. So that it will not be overwritten during the windows update. Here is my system partition -

Number	Start	End	Size	File	Name	Flags
1	1049kB	106MB	105MB	fat32	EFI system partition	boot,esp
2	106MB	123MB	16.8MB		Microsoft reserved partition	msftres
3	123MB	246GB	246GB	ntfs	Basic data partition	msftdata
4	500GB	500GB	556MB	ntfs		hidden, diag
5	246GB	247GB	1074MB	fat32	EFI System Partition	boot,esp
6	247GB	248GB	1074MB	ext4	Swap partition	swap
7	248GB	500GB	252GB	ext4	Linux root	

5,6,7 are for linux.

-- While installing select Microsoft 104/105 key board layout or else it gives error in future

-- In core installation, only kernel, wpa supplicant and other basic GNU tools will be installed.

-- First install xorg-server, xserver-xorg-video-intel, xorg-fonts, libx11-dev libxft-dev libxinerama-dev libxrandr-dev libxkbcommon-dev build-essential

-- I prefer window manager instead of desktop environment. So I suggest you to install following

- st - Minimalist terminal
- dmenu - application launcher
- dwm - Tiling window manager
- herbe - notification app
- scrot - screenshot app
- xpdf - pdf reader
- mocp - minimalist music player
- bluez bluez-tools - bluetooth
- mpv - video player
- redshift - blue light filter
- xbacklight - screen brightness
- go-mtpfs - mobile access
- xfe - file manager (optional)

-- Basic configuration files

##
=> ~/.xinitrc

```
#!/usr/bin/bash

# Set LED backlight
xbacklight -set 5

# Correction for Java applications
wmname LG3D 2
export AWT_TOOLKIT=MToolkit

# Script for monitoring and displaying battery, volume and time information
while true; do
        CHR=$(cat /sys/class/power_supply/BAT0/status)

        if [ "$CHR" = "Discharging" ]
        then
                herbe "BATTERY IS DOWN"
```

```
        fi

        CLK=$( date +'%I:%M')
        volume=$(amixer get Master | grep '%' | head -n 1 | cut -d '[' -f 2 |
cut -d '%' -f 1)
        xsetroot -name "| vol:$volume% | $CLK  "
        sleep 2
done &

redshift -P -O 3000 && sleep 1

# Desktop background
xsetroot -solid '#000000'

# xset dpms 300 400 500; # To trun off display after 5 minutes inactive
# xss-lock -- sh -c 'echo mem | sudo tee /sys/power/state'
# 2>/dev/null &   # To suspend after 10 minutes

exec dwm
```

```
################################################################################
```

~/.aliases

```
# custom commands
alias ..='cd ..'
alias rm="rm -i"
alias x="exit"
alias re="sudo reboot"
alias off="sudo poweroff"
alias l="/bin/ls -1"
alias ll="/bin/ls -la"
alias mount="sudo mount -o umask=000 /dev/sdb1 /mnt/USB"
alias umount="sudo umount /dev/sdb1"
alias install="sudo apt install"
alias update="sudo apt update"
alias upgrade="sudo apt upgrade"
alias remove="sudo apt remove"
alias display_off="xrandr --output eDP1 --off"
alias hdmi="xrandr --output HDMI1 --auto"
alias phone="go-mtpfs ~/MOBILE &"
alias phone_exit="fusermount -u ~/MOBILE"
alias s="startx"
alias -- -="cd -"
alias red="redshift -O 4000"
alias tree="unset LS_COLORS && tree"
alias bt="bluetooth"
alias sp="echo mem | sudo tee /sys/power/state"

# Programs
alias v="vim"
alias xp="xpdf"

# Bookmarks
alias win="cd /mnt/WINDOWS/"
alias movies="cd /mnt/WINDOWS/Users/admin/Videos"
alias d="cd /mnt/WINDOWS/Users/admin/Documents/DATA-1"
alias h="cd /mnt/WINDOWS/Users/admin"
alias dw="cd ~/Downloads"
alias usb="cd /mnt/USB"
```

```
################################################################################
```

~/.bashrc

```
source .aliases
export PS1="\w $:"
export PATH=$PATH:~/.local/bin
```

```
export EDITOR=vim

################################################################################

# ~/.bash_profile

# Load .bashrc for consistency
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# Start X only if we're on the first TTY (ctrl+alt+F1)
if [[ -z $DISPLAY ]] && [[ $(tty) == /dev/tty1 ]]; then
    exec startx
fi

################################################################################
~/.config/redshift/redshift.conf

[redshift]
location-provider=manual

[manual]
lat=0.0
lon=0.0
################################################################################

~/.config/mpv/input.conf

# --- Volume ---
UP      add volume 5        # increase volume
DOWN    add volume -5       # decrease volume
m       cycle mute           # toggle mute

# --- Subtitle ---
s     cycle sub            # cycle subtitle tracks
v     cycle sub-visibility # toggle subtitles on/off

# --- Audio track ---
a     cycle audio          # cycle audio tracks

# --- Playback ---
SPACE     cycle pause          # pause / play
f         cycle fullscreen     # toggle fullscreen
q         quit                 # quit mpv

################################################################################

!/.config/mpv/mpv.conf

# to disable auto turnoff display

stop-screensaver=yes

################################################################################

Custom script for wifi connection

$ net <name>

~/.local/bin/net
#!/bin/bash
# Simple Wi-Fi switcher for home/office
# Requires: wpa_supplicant, dhclient

IFACE="wlan0"

# clear previous records
```

```
reset_iface() {
    sudo killall wpa_supplicant dhclient 2>/dev/null
    sudo ip link set $IFACE down
    sleep 1
    sudo ip link set $IFACE up
    sleep 1
}

case "$1" in

moto)
echo "[*] Connecting to Moto WiFi..."
    reset_iface
    sudo wpa_supplicant -B -i $IFACE -c
/etc/wpa_supplicant/wpa_supplicant-moto.conf
    sudo dhclient $IFACE
    ;;

vivo)
echo "[*] Connecting to Vivo WiFi..."
    reset_iface
    sudo wpa_supplicant -B -i $IFACE -c
/etc/wpa_supplicant/wpa_supplicant-vivo.conf
    sudo dhclient $IFACE
    ;;

cable)
echo "[*] Connecting to Cable WiFi..."
    reset_iface
    sudo wpa_supplicant -B -i $IFACE -c
/etc/wpa_supplicant/wpa_supplicant-cable.conf
    sudo dhclient $IFACE
    ;;
*)
echo "Usage: wifi {moto|vivo|cable}"
;;
esac

##############################################################################

In /etc/wpa_supplicant/wpa_supplicant-name.conf

ctrl_interface=/run/wpa_supplicant
network={
    ssid="xyz"
    psk="password"
}

##############################################################################

custom script for bluetooth connection
~/.local/bin/bt

#!/bin/bash
# Connect to bluetooth device using bluetoothctl

DEVICE="8C:8E:40:C0:AF:00"

bluetoothctl<<EOF
power on
remove $DEVICE
sleep 1
agent on
default-agent
scan on
sleep 2
pair $DEVICE
sleep 2
```

```
trust $DEVICE
sleep 1
connect $DEVICE
sleep 2
quit
EOF

################################################################################

Config file for theme and single column layout for moc

Theme =  black_theme
 Layout1 = playlist(50%,50%,50%,50%)
 Layout2 = ""
 Layout3 = ""

################################################################################

Grub configuration for simple layout -
-- sudo chmod -x 05_debian_theme 20_memtest86+ 30_uefi-firmware
-- Edit menu names in /boot/grub/grub.cfg
-- Edit /etc/default/grub file

GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`grep PRETTY_NAME /etc/lsb-release | cut -d= -f2 | cut -d\" -f2
2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet selinux=0"
GRUB_CMDLINE_LINUX=""
GRUB_TERMINAL=console
GRUB_DISABLE_SUBMENU=y

-- Run sudo update-grub

################################################################################

To acess phone memory
-- install go-mtpfs, libmtp
-- create a folder preferably in home directory for mounting
-- Use alias - phone_mount and phone_umount
-- Create/update /etc/udev/rules.d/51-android.rules

SUBSYSTEM=="usb", ATTR{idVendor}=="22b8", ATTR{idProduct}=="2e82", MODE="0666",
GROUP="plugdev"

################################################################################

To make xbacklight working
-- Create a file in /etc/X11/xorg.conf.d/20-video.conf
-- Write
Section "Device"
    Identifier  "Intel Graphics"
    Driver      "intel"
    Option      "Backlight"  "intel_backlight"
EndSection

################################################################################

To enable tap on click for touchpad
-- install xserver-xorg-input-synaptics
-- Edit/create file /etc/X11/xorg.conf.d/synaptic.conf
Section "InputClass"
        Identifier      "Touchpad"                      # required
        MatchIsTouchpad "yes"                           # required
        Driver          "synaptics"                     # required
        Option          "MinSpeed"              "0.5"
        Option          "MaxSpeed"              "1.0"
        Option          "AccelFactor"           "0.075"
```

```
        Option          "TapButton1"            "1"
        Option          "TapButton2"            "2"     # multitouch
        Option          "TapButton3"            "3"     # multitouch
        Option          "VertTwoFingerScroll"   "1"     # multitouch
        Option          "HorizTwoFingerScroll"  "1"     # multitouch
        Option          "VertEdgeScroll"        "1"
        Option          "CoastingSpeed"         "8"
        Option          "CornerCoasting"        "1"
        Option          "CircularScrolling"     "1"
        Option          "CircScrollTrigger"     "7"
        Option          "EdgeMotionUseAlways"   "1"
        Option          "LBCornerButton"        "8"     # browser "back" btn
        Option          "RBCornerButton"        "9"     # browser "forward" btn
EndSection

###############################################################################

For mounting USB when plugged and Windows partition during boot

-- Use alias usb_mount/umount
alias usb_mount="sudo mount -o umask=000 /dev/sdb1 /mnt/USB"
alias usb_umount="sudo umount /dev/sdb1"
-- Edit /etc/fstab
UUID=XXXX /mnt/WINDOWS ntfs-3g defaults,noatime,uid=1000,gid=1000,umask=022 0 0

###############################################################################

Keep only following daemons in run level 2 (run sysv-rc-conf)

--bluetooth
--dbus
--pulseaudio
--rc.local
--seatd
--sudo
--tlp

###############################################################################

dwm setup file - config.h

/* appearance */
static const unsigned int borderpx  = 1;        /* border pixel of windows */
static const unsigned int snap      = 32;       /* snap pixel */
static const int showbar            = 1;        /* 0 means no bar */
static const int topbar             = 0;        /* 0 means bottom bar */
static const char *fonts[]          = { "ubuntu mono:size=12" };
static const char dmenufont[]       = "ubuntu mono:size=12";
static const char col_gray1[]       = "#222222";
static const char col_gray2[]       = "#444444";
static const char col_gray3[]       = "#bbbbbb";
static const char col_gray4[]       = "#eeeeee";
static const char col_cyan[]        = "#005577";
static const char *colors[][3]      = {
        /*               fg         bg         border   */
        [SchemeNorm] = { col_gray3, col_gray1, col_gray2 },
        [SchemeSel]  = { col_gray4, col_cyan,  col_cyan  },
};

/* tagging */
static const char *tags[] = { "1", "2", "3", "4", "5", "6" };

static const Rule rules[] = {
        /* xprop(1):
         *      WM_CLASS(STRING) = instance, class
         *      WM_NAME(STRING) = title
         */
        /* class      instance    title       tags mask     isfloating   monitor
```

```
 * */
        { "Gimp",     NULL,       NULL,       0,            1,           -1 },
        { "Firefox",  NULL,       NULL,       1 << 8,       0,           -1 },
};

/* layout(s) */
static const float mfact     = 0.55; /* factor of master area size [0.05..0.95]
*/
static const int nmaster     = 1;    /* number of clients in master area */
static const int resizehints = 1;    /* 1 means respect size hints in tiled
resizals */
static const int lockfullscreen = 1; /* 1 will force focus on the fullscreen
window */

static const Layout layouts[] = {
        /* symbol     arrange function */
        { "[]=",      tile },    /* first entry is default */
        { "><>",      NULL },    /* no layout function means floating behavior
*/
        { "[M]",      monocle },
};

/* key definitions */
#define MODKEY Mod1Mask
#define TAGKEYS(KEY,TAG) \
        { MODKEY,                       KEY,      view,           {.ui = 1 <<
TAG} }, \
        { MODKEY|ControlMask,           KEY,      toggleview,     {.ui = 1 <<
TAG} }, \
        { MODKEY|ShiftMask,             KEY,      tag,            {.ui = 1 <<
TAG} }, \
        { MODKEY|ControlMask|ShiftMask, KEY,      toggletag,      {.ui = 1 <<
TAG} },

/* helper for spawning shell commands in the pre dwm-5.0 fashion */
#define SHCMD(cmd) { .v = (const char*[]){ "/bin/sh", "-c", cmd, NULL } }
/* commands */
static char dmenumon[2] = "0"; /* component of dmenucmd, manipulated in spawn()
*/
static const char *dmenucmd[] = { "dmenu_run", "-m", dmenumon, "-fn", dmenufont,
"-nb", col_gray1, "-nf", col_gray3, "-sb", col_cyan, "-sf", col_gray4, NULL };
static const char *termcmd[]  = { "st", NULL };
static const char *browsercmd[]= {"chromium",NULL};
static const char *fmcmd[] = { "xfe", NULL };
static const char *htopcmd[] = { "st", "-e", "htop", NULL };
static const char *mocpcmd[] = { "st", "-e", "mocp", NULL };
static const char *volup[] = { "amixer", "set", "Master", "5%+", NULL };
static const char *voldown[] = { "amixer", "set", "Master", "5%-", NULL };
static const char *brtup[] = { "xbacklight", "-inc","5", NULL };
static const char *brtdown[] = { "xbacklight", "-dec","5", NULL };
static const char *lock[] = { "st", "-e", "slock" };
static const char *scrotcmd[]  = { "scrot", "-t", "25", NULL };
static const char *scrotfocusedcmd[]  = { "scrot", "--focused", NULL };

static Key keys[] = {
        /* modifier                     key        function        argument */
        { MODKEY,                       XK_d,      spawn,          {.v =
dmenucmd } },
        { MODKEY          ,             XK_Return, spawn,          {.v = termcmd
} },
        { MODKEY,                       XK_b,      togglebar,      {0} },
        { MODKEY,                       XK_j,      focusstack,     {.i = +1 } },
        { MODKEY,                       XK_k,      focusstack,     {.i = -1 } },
        { MODKEY,                       XK_w,      incnmaster,     {.i = +1 } },
        { MODKEY|ShiftMask,             XK_w,      incnmaster,     {.i = -1 } },
        { MODKEY,                       XK_h,      setmfact,       {.f = -0.05}
},
        { MODKEY,                       XK_l,      setmfact,       {.f = +0.05}
```

```
},
        { MODKEY|ShiftMask,             XK_Return, zoom,           {0} },
        { MODKEY,                       XK_Tab,    view,           {0} },
        { MODKEY,                       XK_q,      killclient,     {0} },
        { MODKEY,                       XK_t,      setlayout,      {.v =
&layouts[0]} },
        { MODKEY,                       XK_m,      setlayout,      {.v =
&layouts[1]} },
        { MODKEY,                       XK_f,      setlayout,      {.v =
&layouts[2]} },
        { MODKEY,                       XK_space,  setlayout,      {0} },
        { MODKEY|ShiftMask,             XK_space,  togglefloating, {0} },
        { MODKEY,                       XK_0,      view,           {.ui = ~0 }
},
        { MODKEY|ShiftMask,             XK_0,      tag,            {.ui = ~0 }
},
        { MODKEY,                       XK_comma,  focusmon,       {.i = -1 } },
        { MODKEY,                       XK_period, focusmon,       {.i = +1 } },
        { MODKEY|ShiftMask,             XK_comma,  tagmon,         {.i = -1 } },
        { MODKEY|ShiftMask,             XK_period, tagmon,         {.i = +1 } },
        { MODKEY,                       XK_s,      spawn,          SHCMD("echo
mem | sudo tee /sys/power/state") },
        { MODKEY,                       XK_i,            spawn, {.v = browsercmd
} },
        { MODKEY|ShiftMask,             XK_x,            spawn, {.v = lock } },
        { MODKEY,                       XK_p,            spawn, {.v = fmcmd } },
        { MODKEY,                       XK_o,            spawn, {.v = htopcmd }
},
        { MODKEY,                       XK_z,            spawn, {.v = mocpcmd }
},
        { MODKEY,                       XK_Up,           spawn, {.v = volup } },
        { MODKEY,                       XK_Down,         spawn, {.v = voldown }
},
        { MODKEY,                       XK_8,           spawn, {.v = brtup } },
        { MODKEY,                       XK_9,           spawn, {.v = brtdown }
},
        TAGKEYS(                        XK_1,                      0)
        TAGKEYS(                        XK_2,                      1)
        TAGKEYS(                        XK_3,                      2)
        TAGKEYS(                        XK_4,                      3)
        TAGKEYS(                        XK_5,                      4)
        TAGKEYS(                        XK_6,                      5)
        { MODKEY|ShiftMask,             XK_e,      quit,           {0} },
        { 0,            XK_Print,   spawn,      {.v = scrotcmd } },
        { ShiftMask,    XK_Print,   spawn,      {.v = scrotfocusedcmd } },
        { ControlMask,  XK_Print,   spawn,      SHCMD("sleep 1s;scrot --select")
},

};


/* button definitions */
/* click can be ClkTagBar, ClkLtSymbol, ClkStatusText, ClkWinTitle,
 * ClkClientWin, or ClkRootWin */
static const Button buttons[] = {
        /* click                event mask      button          function
 * argument */
        { ClkLtSymbol,          0,              Button1,        setlayout,
{0} },
        { ClkLtSymbol,          0,              Button3,        setlayout,
{.v = &layouts[2]} },
        { ClkWinTitle,          0,              Button2,        zoom,
{0} },
        { ClkStatusText,        0,              Button2,        spawn,
{.v = termcmd } },
        { ClkClientWin,         MODKEY,         Button1,        movemouse,
{0} },
        { ClkClientWin,         MODKEY,         Button2,        togglefloating,
```

```
{0} },
        { ClkClientWin,         MODKEY,         Button3,        resizemouse,
{0} },
        { ClkTagBar,            0,              Button1,        view,
{0} },
        { ClkTagBar,            0,              Button3,        toggleview,
{0} },
        { ClkTagBar,            MODKEY,         Button1,        tag,
{0} },
        { ClkTagBar,            MODKEY,         Button3,        toggletag,
{0} },
};

################################################################################
```