**1. index.php**

<?php

$DEBUG = False;

session\_start();

if($\_SESSION["login"] == False)

{

header("Location: login.php");

die();

}

$kernel = shell\_exec("uname -r");

$uptime = shell\_exec("uptime -p | cut -c 3-");

$wlan\_ip = shell\_exec('ifconfig wlan0 | grep -oP "inet \d+\.\d+\.\d+\.\d+" | cut -c 6-');

$version = shell\_exec('ipmi version');

$host\_ip = shell\_exec('cat /home/ipmi/.ipmi/ipmi.conf | grep "HostIP=" | cut -c 8-');

$host\_power = shell\_exec("ipmi status | cut -c 15-");

$host\_shell = shell\_exec("ipmi ping");

?>

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="bootstrap.css">

<!-- custom css -->

<link rel="stylesheet" href="style.css">

<title>G-IPMI</title>

</head>

<body>

<!-- Navigation navbar-dark bg-dark -->

<nav class="navbar navbar-expand-lg fixed-top navbar-dark">

<div class="container">

<a class="navbar-brand" href="index.php">Generic IPMI web portal</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarResponsive">

<ul class="navbar-nav ml-auto">

<li class="nav-item active">

<a class="nav-link" href="index.php">Control panel

<span class="sr-only">(current)</span>

</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false" id="dropdown01">Controls</a>

<div class="dropdown-menu" aria-labelledby="dropdown01">

<a class="dropdown-item" href="ipmi-bridge.php?arg=start">Start</a>

<a class="dropdown-item" href="ipmi-bridge.php?arg=stop">Stop</a>

<a class="dropdown-item" href="ipmi-bridge.php?arg=restart">Restart</a>

</div>

</li>

<li class="nav-item">

<a class="nav-link" href="https://github.com/ChipTechno/IPMI-card">Repository</a>

</li>

<li class="nav-item">

<a class="nav-link" href="logout.php">Logout</a>

</li>

</ul>

</div>

</div>

</nav>

<!-- main content -->

<div class="container main-container">

<div class="row main-content">

<div class="col-md-12">

<p class="text-white bg-primary p-3 mb-2 h4">System overview</p>

<table class="table table-condensed table-borderless">

<tbody>

<tr class="d-flex">

<th scope="row" class="text-nowrap col-5">Power state</th>

<td class="col-7 text-nowrap"><?php echo $host\_power; ?></td>

</tr>

<tr class="d-flex">

<th scope="row" class="text-nowrap col-5">SSH</th>

<td class="col-7 text-nowrap"><?php echo $host\_shell; ?></td>

</tr>

<tr class="d-flex">

<th scope="row" class="text-nowrap col-5">IP address</th>

<td class="col-7 text-nowrap"><?php echo $host\_ip; ?></td>

</tr>

</tbody>

</table>

</div>

<div class="col-md-12">

<p class="text-white bg-secondary p-3 mb-2 h4">IPMI</p>

<table class="table table-condensed table-borderless">

<tbody>

<tr class="d-flex">

<th scope="row" class="text-nowrap col-5">Kernel</th>

<td class="col-7 text-nowrap"><?php echo $kernel; ?></td>

</tr>

<tr class="d-flex">

<th scope="row" class="text-nowrap col-5">Uptime</th>

<td class="col-7 text-nowrap"><?php echo $uptime; ?></td>

</tr>

<tr class="d-flex">

<th scope="row" class="text-nowrap col-5">IP address</th>

<td class="col-7 text-nowrap"><?php echo $wlan\_ip; ?></td>

</tr>

<tr class="d-flex">

<th scope="row" class="text-nowrap col-5 text-truncate">Software</th>

<td class="col-7 text-nowrap"><?php echo $version; ?></td>

</tr>

</tbody>

</table>

</div>

</div>

</div>

<!-- footer bg-dark -->

<footer class="fixed-bottom text-center py-3 navbar-dark">

<div class="container">

<p class="m-0 text-white">Generic Intelligent Platform Management Interface</p>

<a href="https://github.com/ChipTechno/IPMI-card">https://github.com/ChipTechno/IPMI-card</a>

</div>

</footer>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="jquery-3.3.1.slim.js"></script>

<script src="popper.js"></script>

<script src="bootstrap.js"></script>

</body>

</html>

**2. ipmi-bridge.php**

<?php

$DEBUG = True;

session\_start();

if($\_SESSION["login"] == False)

{

header("Location: login.php");

die();

}

$arg = $\_GET["arg"];

if($arg == "start")

shell\_exec("ipmi start");

if($arg == "stop")

shell\_exec("ipmi stop");

if($arg == "restart")

shell\_exec("ipmi restart");

header("Location: msg.php?msg=Please wait...");

?>

**3. login\_action.php**

<?php

$DEBUG = True;

$usrname = $\_GET["usrname"];

$passwd = $\_GET["passwd"];

if($DEBUG)

echo $usrname."@".$passwd."<br/>";

session\_start();

//calc hash

$usrHash = hash('sha256', $usrname);

$passwdHash = hash('sha256', $passwd);

//read hash from config file

//cat ipmi.conf | grep "WebUser=" | cut -c 9-

//cat ipmi.conf | grep "WebPasswd=" | cut -c 11-

$savedUsrHash = shell\_exec('cat /home/ipmi/.ipmi/ipmi.conf | grep "WebUser=" | cut -c 9-');

$savedPasswdHash = shell\_exec('cat /home/ipmi/.ipmi/ipmi.conf | grep "WebPasswd=" | cut -c 11-');

//trim the hash string

if(trim($usrHash) == trim($savedUsrHash) && trim($passwdHash) == trim($savedPasswdHash))

{

$\_SESSION["login"] = True;

header("Location: index.php");

}

else

{

$\_SESSION["login"] = False;

header("Location: msg.php?msg=Wrong username or password!");

}

?>

**4. login.php**

<?php

session\_start();

if($\_SESSION["login"] == True)

{

header("Location: index.php");

die();

}

?>

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="bootstrap.css">

<!-- custom css -->

<link rel="stylesheet" href="style.css">

<title>G-IPMI</title>

</head>

<body>

<!-- Navigation navbar-dark bg-dark -->

<nav class="navbar navbar-expand-lg fixed-top navbar-dark">

<div class="container">

<a class="navbar-brand" href="index.php">Generic IPMI web portal</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarResponsive">

<ul class="navbar-nav ml-auto">

<li class="nav-item active">

<a class="nav-link" href="index.php">Control panel

<span class="sr-only">(current)</span>

</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false" id="dropdown01">Controls</a>

<div class="dropdown-menu" aria-labelledby="dropdown01">

<a class="dropdown-item" href="ipmi-bridge.php?arg=start">Start</a>

<a class="dropdown-item" href="ipmi-bridge.php?arg=stop">Stop</a>

<a class="dropdown-item" href="ipmi-bridge.php?arg=restart">Restart</a>

</div>

</li>

<li class="nav-item">

<a class="nav-link" href="https://github.com/ChipTechno/IPMI-card">Repository</a>

</li>

<li class="nav-item">

<a class="nav-link" href="logout.php">Logout</a>

</li>

</ul>

</div>

</div>

</nav>

<!-- main content -->

<div class="container main-container">

<div class="row main-content">

<div class="col-md-12">

<form action="/login\_action.php">

<div class="form-group">

<label for="exampleInputEmail1">Username</label>

<input type="text" class="form-control" id="exampleInputEmail1" placeholder="Enter username" name="usrname" required>

</div>

<div class="form-group">

<label for="exampleInputPassword1">Password</label>

<input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password" name="passwd" required>

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

</div>

</div>

<!-- footer bg-dark -->

<footer class="fixed-bottom text-center py-3 navbar-dark">

<div class="container">

<p class="m-0 text-white">Generic Intelligent Platform Management Interface</p>

<a href="https://github.com/ChipTechno/IPMI-card">https://github.com/ChipTechno/IPMI-card</a>

</div>

</footer>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="jquery-3.3.1.slim.js"></script>

<script src="popper.js"></script>

<script src="bootstrap.js"></script>

</body>

</html>

**5. logout.php**

<?php

session\_start();

$\_SESSION["login"] = False;

//unset($\_SESSION["login"]);

header("Location: login.php");

die();

?>

**6. msg.php**

<?php

$msg = $\_GET["msg"];

header("refresh:3;url=index.php");

?>

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="bootstrap.css">

<!-- custom css -->

<link rel="stylesheet" href="style.css">

<title>G-IPMI</title>

</head>

<body>

<!-- Navigation navbar-dark bg-dark -->

<nav class="navbar navbar-expand-lg fixed-top navbar-dark">

<div class="container">

<a class="navbar-brand" href="index.php">Generic IPMI web portal</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarResponsive">

<ul class="navbar-nav ml-auto">

<li class="nav-item active">

<a class="nav-link" href="index.php">Control panel

<span class="sr-only">(current)</span>

</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false" id="dropdown01">Controls</a>

<div class="dropdown-menu" aria-labelledby="dropdown01">

<a class="dropdown-item" href="ipmi-bridge.php?arg=start">Start</a>

<a class="dropdown-item" href="ipmi-bridge.php?arg=stop">Stop</a>

<a class="dropdown-item" href="ipmi-bridge.php?arg=restart">Restart</a>

</div>

</li>

<li class="nav-item">

<a class="nav-link" href="https://github.com/ChipTechno/IPMI-card">Repository</a>

</li>

<li class="nav-item">

<a class="nav-link" href="logout.php">Logout</a>

</li>

</ul>

</div>

</div>

</nav>

<!-- main content -->

<div class="container main-container">

<div class="row main-content">

<div class="col-md-12">

<h4 class="display-4 text-center mt-5"><?php echo $msg; ?></h4><br/>

</div>

</div>

</div>

<!-- footer bg-dark -->

<footer class="fixed-bottom text-center py-3 navbar-dark">

<div class="container">

<p class="m-0 text-white">Generic Intelligent Platform Management Interface</p>

<a href="https://github.com/ChipTechno/IPMI-card">https://github.com/ChipTechno/IPMI-card</a>

</div>

</footer>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="jquery-3.3.1.slim.js"></script>

<script src="popper.js"></script>

<script src="bootstrap.js"></script>

</body>

</html>

**7. style.css**

.main-content

{

margin-top : 10vh;

}

.navbar-dark

{

background-color:#1e1e1e;

}

.custom-footer-dark

{

background-color:#1e1e1e;

}

.custom-footer-dark a

{

color:#5396cd;

}

.custom-footer-dark a:hover

{

color:#5396cd;

}

body

{

background-color:#2d2d2d;

color:#d3d3d3;

}

th

{

color:#ff7043;

}

td

{

color : #d3d3d3;

}

.navbar .dropdown-menu {

background-color: #1e1e1e;

color:#d3d3d3;

border:none;

}

/\* and this styles the dropdwon trigger link, when open \*/

.navbar .dropdown.show a {

background-color: #1e1e1e;

color:#d3d3d3;

border:none;

}

.navbar .dropdown.show a:hover {

background-color: #1e1e1e;

color:#0087af;

border:none;

}

**8. Ipmi manページ**

.\" Manpage for ipmi.

.\" Contact login721@gmail.com to correct errors or typos.

.TH man 1 "2018-12-31" "1.2 beta" "ipmi man page"

.SH NAME

ipmi \- Intelligent Platform Management Interface for generic motherboard.

.SH SYNOPSIS

ipmi [help]

.br

[version]

.br

[status]

.br

[shell]

.br

[start]

.br

[stop] [-f]

.br

[restart] [-f]

.br

[setup]

.SH DESCRIPTION

ipmi is shell command line to control IPMI module.

.SH OPTIONS

.TP

\fBhelp\fP

.br

Print a short help text and exit.

.TP

\fBversion\fP

.br

Display application version.

.TP

\fBstatus\fP

.br

Display the host system's current status.

.TP

\fBshell\fP

.br

Connect to the host system via ssh.

.TP

\fBstart\fP

.br

Start the host system.

.TP

\fBstop\fP

.br

Stop(shutdown) the host system.

.TP

\fBrestart\fP

.br

Restart the host system.

.TP

\fB-f\fP

.br

Force stop or restart the host system.

.TP

\fBsetup\fP

.br

Setup ip address of host system and user/password for web interface.

.SH BUGS

No known bugs.

.SH AUTHOR

ChipCE (login721@gmail.com) <https://github.com/ChipTechno>

**9. Neofetch config**

# Neofetch config file

print\_info() {

info line\_break

info line\_break

info line\_break

info line\_break

prin "Generic Intelligent Platform Management Interface"

info title

info underline

info "OS" distro

info "Host" model

info "Kernel" kernel

info "Uptime" uptime

info "Shell" shell

info "CPU" cpu

info "GPU" gpu

info "Memory" memory

info "Local IP" local\_ip

info "Public IP" public\_ip

info line\_break

info line\_break

info line\_break

info line\_break

info line\_break

info line\_break

info line\_break

}

kernel\_shorthand="on"

distro\_shorthand="off"

os\_arch="on"

uptime\_shorthand="on"

shell\_path="off"

shell\_version="on"

speed\_type="bios\_limit"

speed\_shorthand="off"

cpu\_brand="on"

cpu\_speed="on"

cpu\_cores="logical"

cpu\_temp="off"

gpu\_brand="on"

gpu\_type="all"

refresh\_rate="off"

gtk\_shorthand="off"

gtk2="on"

gtk3="on"

public\_ip\_host="http://ident.me"

disk\_show=('/')

disk\_subtitle="mount"

music\_player="auto"

song\_shorthand="off"

install\_time="on"

install\_time\_format="12h"

colors=(distro)

bold="on"

underline\_enabled="on"

underline\_char="-"

block\_range=(0 7)

color\_blocks="on"

block\_width=3

block\_height=1

bar\_char\_elapsed="-"

bar\_char\_total="="

bar\_border="on"

bar\_length=15

bar\_color\_elapsed="distro"

bar\_color\_total="distro"

cpu\_display="off"

memory\_display="off"

battery\_display="off"

disk\_display="off"

image\_backend="ascii"

image\_source="auto"

ascii\_distro="auto"

ascii\_colors=(distro)

ascii\_bold="on"

image\_loop="off"

thumbnail\_dir="${XDG\_CACHE\_HOME:-${HOME}/.cache}/thumbnails/neofetch"

crop\_mode="normal"

crop\_offset="center"

image\_size="auto"

gap=3

yoffset=0

xoffset=0

background\_color=

scrot="off"

scrot\_cmd="auto"

scrot\_name="neofetch-$(date +%F-%I-%M-%S-${RANDOM}).png"

image\_host="teknik"

stdout="off"

config\_version="3.4.0"

**10. neofetch**

${c1}10001111000111010111101${c2}

${c1}00101010010001110101011${c2}

${c1}10101010010100000110100${c2}

${c1}00100001${c2}

${c1}10001001${c2}

${c1}11010111${c2} ${c1}11111${c2} ${c1}1101${c2} ${c1}1${c2} ${c1}0${c2} ${c1}10110${c2}

${c1}01011100${c2} ${c1}0${c2} ${c1}1${c2} ${c1}0${c2} ${c1}10${c2} ${c1}10${c2} ${c1}1${c2}

${c1}01111000${c2} ${c1}1${c2} ${c1}0100${c2} ${c1}0${c2} ${c1}0${c2} ${c1}0${c2} ${c1}1${c2}

${c1}01111100${c2} ${c1}1${c2} ${c1}1${c2} ${c1}0${c2} ${c1}1${c2} ${c1}1${c2}

${c1}01011111${c2} ${c1}00110${c2} ${c1}1${c2} ${c1}0${c2} ${c1}0${c2} ${c1}11100${c2}

${c1}00000111${c2} ${c1}00101010${c2}

${c1}10011001${c2} ${c1}00011100${c2}

${c1}10000010${c2} ${c1}00001000${c2}

${c1}00110011110100111101110010000000001101${c2}

${c1}01000110001011011111011${c2}

${c1}01101000000101000000110${c2}

${c1}10001000011100100111001${c2}

**11. Ipmi config**

HostIP=192.168.1.1

WebUser=d7cea7305a333d5c9cebd7891abbee78960632bbb1fa5dd24122b940ff823ace

WebPasswd=d7cea7305a333d5c9cebd7891abbee78960632bbb1fa5dd24122b940ff823ace

**12. Ipmi.py**

import sys

import os

import RPi.GPIO as gpio

import time

gpio.setwarnings(False)

gpio.setmode(gpio.BCM)

# power monitor (active low)

gpio.setup(4, gpio.IN)

# relay control (active high)

gpio.setup(18, gpio.OUT)

# function define ---------------------------------------

# shutdown control

def stop(\_force):

#print("Stop")

# if the sys is already off

if powerState() == False:

print("System state is OFF.")

return

# if not , then turn it off

if \_force == True:

print("Trying to force stop...")

gpio.output(18, gpio.HIGH)

while powerState():

pass

gpio.output(18, gpio.LOW)

return

else:

# just click the button

print("Trying in to stop...")

gpio.output(18, gpio.HIGH)

time.sleep(0.3)

gpio.output(18, gpio.LOW)

return

# power-on control

def start():

if powerState() == True:

print("System state is ON.")

return

else:

print("Trying in to start...")

gpio.output(18, gpio.HIGH)

time.sleep(0.3)

gpio.output(18, gpio.LOW)

return

# restart control

def restart(\_force):

print("Trying to restart...")

print("Step 1 of 2 : Stop...")

stop(\_force)

while powerState() == True:

pass

time.sleep(1)

print("Step 2 of 2 : Start..")

start()

return

# status report

def status():

#print("Status report")

# power

if powerState():

print("Power state : ON")

else:

print("Power state : OFF")

return

# get power-state

def powerState():

if(gpio.input(4) == False):

return True

else:

return False

def default():

gpio.output(18, gpio.LOW)

# -----------------------------------------------------

# get force and wait flag

force = False

if (len(sys.argv) > 3):

print("Error : Too much arguments!")

gpio.cleanup()

sys.exit()

if (len(sys.argv) <2 ):

print("Error : Too few arguments")

gpio.cleanup()

sys.exit()

if (len(sys.argv) == 3):

if (sys.argv[2] == "-f"):

force = True

if (sys.argv[1] == "start"):

start()

gpio.cleanup()

sys.exit()

if (sys.argv[1] == "stop"):

stop(force)

gpio.cleanup()

sys.exit()

if (sys.argv[1] == "restart"):

restart(force)

gpio.cleanup()

sys.exit()

if (sys.argv[1] == "status"):

status()

gpio.cleanup()

sys.exit()

print("Error : Unknow arguments")

gpio.cleanup()

sys.exit()

**13. ipmi.sh**

#!/bin/bash

# var

version="1.2 beta"

# function

ipmi-help () {

# or just call "man ipmi" here :3

echo "Generic IPMI std $version"

echo -e "USAGE \n\t ipmi [help]"

echo -e "\t [version]"

echo -e "\t [status]"

echo -e "\t [shell]"

echo -e "\t [start]"

echo -e "\t [stop] [-f]"

echo -e "\t [restart] [-f]"

echo -e "\t [setup]"

echo "OPTIONS"

echo -e "\t help \n\t\t Print a short help text and exit."

echo -e "\t version \n\t\t Display application version."

echo -e "\t status \n\t\t Display the host system's current status."

echo -e "\t shell \n\t\t Connect to the host system via ssh."

echo -e "\t start \n\t\t Start the host system."

echo -e "\t stop \n\t\t Stop(shutdown) the host system."

echo -e "\t restart \n\t\t restart the host system."

echo -e "\t -f \n\t\t Force stop or restart the host system."

echo -e "\t setup \n\t\t Setup IPMI."

}

# check if the first arg is empty

if [ -z "$1" ]; then

ipmi-help

exit 0

fi

# force flag

\_force=false

# check 2nd arg

if [ -n "$2" ]; then

if [ "$2" == "-f" ]; then

\_force=true

else

echo -e "Error : Unknown argument \"$2\" !"

exit 1

fi

fi

# open ssh connection

if [ "$1" = "shell" ]; then

if [ $# -gt 1 ]; then

echo "Error : Too much arguments!"

exit 1

fi

# read config file

host=(`cat /home/ipmi/.ipmi/ipmi.conf | grep "HostIP=" | cut -c 8-`)

echo "$host"

if [ "$host" = "" ]; then

echo "Cannot get host IP address!"

else

printf "Username : "

read username

ssh $username@$host

fi

exit 0

fi

# handle status report

if [ "$1" = "status" ]; then

if [ $# -gt 1 ]; then

echo "Error : Too much arguments!"

exit 1

fi

# run status report script

python /home/ipmi/.ipmi/ipmi.py status

exit 0

fi

# handle start

if [ "$1" = "start" ]; then

if [ $# -gt 1 ]; then

echo "Error : Too much arguments!"

exit 1

fi

echo "Start : python ipmi.py start"

python /home/ipmi/.ipmi/ipmi.py start

exit 0

fi

# handle stop

if [ "$1" = "stop" ]; then

if [ $# -gt 2 ]; then

echo "Error : Too much arguments!"

exit 1

fi

\_command="python /home/ipmi/.ipmi/ipmi.py stop"

if [ "$\_force" == true ]; then

\_command="$\_command -f"

fi

echo "Stop : $\_command"

$\_command

exit 0

fi

# handle restart

if [ "$1" = "restart" ]; then

if [ $# -gt 2 ]; then

echo "Error : Too much arguments!"

exit 1

fi

\_command="python /home/ipmi/.ipmi/ipmi.py restart"

if [ "$\_force" == true ]; then

\_command="$\_command -f"

fi

echo "Restart : $\_command"

$\_command

exit 0

fi

# handle setup

if [ "$1" = "setup" ]; then

if [ $# -gt 1 ]; then

echo "Error : Too much arguments!"

exit 1

fi

# delete old config line

if [ -f /home/ipmi/.ipmi/ipmi.conf ]; then

echo "Delete old config file."

rm /home/ipmi/.ipmi/ipmi.conf

fi

read -p "Enter ip address of the host PC : " hostIP

echo "HostIP=$hostIP" >> /home/ipmi/.ipmi/ipmi.conf

echo ""

read -p "Enter username for web interface : " webUser

userHash=($(echo -n $webUser | sha256sum | cut -c -64))

echo "WebUser=$userHash" >> /home/ipmi/.ipmi/ipmi.conf

echo ""

read -p "Enter password for web interface : " webPasswd

passwdHash=($(echo -n $webPasswd | sha256sum | cut -c -64))

echo "WebPasswd=$passwdHash" >> /home/ipmi/.ipmi/ipmi.conf

echo ""

echo "Done!"

exit 0

fi

# handle version

if [ "$1" = "version" ]; then

if [ $# -gt 1 ]; then

echo "Error : Too much arguments!"

exit 1

fi

echo $version

exit 0

fi

# handle ping

if [ "$1" = "ping" ]; then

if [ $# -gt 1 ]; then

echo "Error : Too much arguments!"

exit 1

fi

host=(`cat /home/ipmi/.ipmi/ipmi.conf | grep "HostIP=" | cut -c 8-`)

if [ "$host" = "" ]; then

echo "Cannot get host IP address!"

else

pingRes=($(ping -q -c1 $host > /dev/null))

if [ $? -eq 0 ]

then

echo "Available"

else

echo "Unavailable"

fi

fi

exit 0

fi

echo -e "Error : Unknown argument \"$1\" !"

exit 1

**14. setup.sh**

#!/bin/bash

# root check

FILE="/tmp/out.$$"

GREP="/bin/grep"

if [[ $EUID -ne 0 ]]; then

echo "This script must be run as root!" 1>&2

exit 1

fi

read -p "Install IPMI control program ? y/n : " uConfirm

if [ "$uConfirm" != "y" ]; then

echo "Exit installer!"

exit 1

fi

# create user

echo "Create user ipmi"

adduser ipmi

# add to sudo group

echo "Add ipmi to sudo group."

usermod -a -G sudo ipmi

# Add ipmi to gpio user group

echo "Add ipmi to gpio group."

sudo usermod -a -G gpio ipmi

# update

echo "Check and isntall updates"

apt-get update

apt-get dist-upgrade

# install package

echo "Install packages..."

apt-get install -y apache2 php libapache2-mod-php neofetch

# Add www-data to gpio user group

echo "Add www-data to gpio group."

sudo usermod -a -G gpio www-data

echo "Enable auto-startup for apache."

#sudo systemctl enable apache2

#sudo systemctl start apache2

sudo systemctl enable apache

sudo systemctl start apache

#echo "Copy nodered script."

echo "Enable auto-startup for nodered."

sudo systemctl enable nodered

sudo systemctl start nodered

# ipmi to /bin

echo "Copy ipmi to /bin/ipmi"

yes | cp -rf prog/ipmi.sh /bin/ipmi

chmod 775 /bin/ipmi

# copy ipmi.conf

echo "Copy ipmi.conf to ~/.ipmi/ipmi.conf"

if [ ! -d "/home/ipmi/.ipmi" ]; then

mkdir /home/ipmi/.ipmi

fi

yes | cp -rf prog/ipmi.conf /home/ipmi/.ipmi/ipmi.conf

# copy ipmi.py

echo "Copy ipmi.py to ~/.ipmi/ipmi.py"

yes | cp -rf prog/ipmi.py /home/ipmi/.ipmi/ipmi.py

chown -R ipmi:ipmi /home/ipmi/.ipmi

# neofetch conf

echo "Copy neofetch config to ~/.config/neofetch/config"

if [ ! -d "/home/ipmi/.config" ]; then

mkdir /home/ipmi/.config

fi

if [ ! -d "/home/ipmi/.config/neofetch" ]; then

mkdir /home/ipmi/.config/neofetch

fi

yes | cp -rf neofetch/config /home/ipmi/.config/neofetch/config

chown -R ipmi:ipmi /home/ipmi/.config

chmod 775 /home/ipmi/.config/neofetch/config

# neofetch ascii

echo "Copy neofetch ascii to /usr/share/neofetch/ascii/distro/raspbian"

yes | cp -rf neofetch/ipmi /usr/share/neofetch/ascii/distro/raspbian

chmod 775 /usr/share/neofetch/ascii/distro/raspbian

# add neofetch bo bashrc

echo "Add neifetch entry to bashrc"

echo "#neofetch" >> /home/ipmi/.bashrc

echo "neofetch" >> /home/ipmi/.bashrc

# man

echo "Copy ipmi to /usr/share/man/man1/ipmi.1"

yes | cp -rf man/ipmi /usr/share/man/man1/ipmi.1

gzip /usr/share/man/man1/ipmi.1

# copy www

echo "copy web interface to /var/www/html"

rm -rf /var/www/html/index.html

yes | cp -rf html /var/www

# copy boot file

# echo "Add custom cmdline.txt to /boot"

# cp

echo "Done!"

exit 0