

Segurança em Sistemas Operacionais e Redes de Computadores I		Professor: Robson Parmezan Bonidias	
Nome: Jeanne Dalio Oliveira Nome: Lais Camarini Moura Nome: Rafael Santos Santini Nome: Ralph Schutez Muraro		RA: 0210971921037 RA: 0210971921035 RA: 0210972113043 RA: 0210971921018	
Curso: Segurança da Informação	Turma: 5º Semestre	Período: Noite	Data: 07SET2021

Atividade - Introdução ao Shell Script

Vence 07SET2021.

INSTRUÇÕES

Os detalhes do trabalho estão nos comentários do script:

SCRIPT - COMANDOS E CONSULTAS EM SHELL

```
#!/bin/bash
#Script trabalho de shell.
#Autores:
#      JEANNE DALIO OLIVEIRA
#      LAIS CAMARINI MOURA
#      RAFAEL SANTOS SANTINI
#      RALPH SCHUTEZ MURARO
#
# *****
# ** Este script tem por finalidade ser uma atividade complementar para atribuição de nota na disciplina Segurança em Sistemas **
# ** Operacionais e Redes de Computadores I **
# ** Professor: ROBSON PARMEZAN BONIDIA **
# **
# ** Esse script foi desenvolvido utilizando ferramentas padrões para execução na maiorias dos sistemas baseados em linux, entre- **
# ** tanto, pode ocorrer a necessidade de alterar ou instalar recursos. **
# *****
#
# QUESTÕES PROBLEMA !!!
# -----
# 1 - Crie um Script que após executado solicite dois valores como entrada. Esses valores
#      devem ser somados e o resultado exibido para o usuário.
# -----
# 2 - Crie um script que gere um relatório de algumas informações da máquina atual:
#
#      a) Nome da Máquina,
#      b) Data e Hora Atual,
#      c) Desde quando a máquina está ativa,
#      d) Versão do Kernel,
#      e) Quantidade de CPUs/Cores,
#      f) Modelo da CPU,
#      g) Total de Memória RAM Disponível,
#      h) Partições.
#
# OBS: O maior desafio desse exercício é combinar os comandos para extrair e formatar a informação
#      desejada.
# DICA 1 - Use os comandos tr, cut, grep, sed e etc para isso.
# DICA 2 - Use as ajudas pelo man e help para identificar a melhor opção em cada parâmetro.
# -----
# 3 - Crie um script que receba um nome de usuário como parâmetro e exiba as seguintes
#      informações:
#
#      a) UID do usuário,
#      b) Nome Completo / Descrição do Usuário, Total em Uso no /home do usuário, Informações do
#      c) último login do usuário.
#      d) Opcional: Validar se o usuário existe ou não.
# -----
#
# dialog --crate-rc $HOME/.dialogrc
# export DIALOGRC=$HOME/tema-verde.cfg
```

Faculdade de Tecnologia de Ourinhos

```
for ((i=1; i<=100; i++));

do
(echo $i; sleep 0.1) | dialog \
--backtitle \
**** Analisador de recursos e usuários **** \
--title '*** Verificando Dados da Maquina ***' \
--gauge '\n\n ==> Varendo informações necessárias para consultas e \nanalisando recursos ...' \
10 60 $i

done

menu () {
m=$(dialog \
--stdout \
--backtitle \
**** Analisador de recursos e usuários **** \
--title '*** MENU OPÇÕES ***' \
" **** MENU OPÇÕES **** " \
--menu \
"\nSelecione uma função" \
0 0 4 \
1 " [1] Adição (Soma)" \
2 " [2] Informações da Maquina" \
3 " [3] Consulta Usuário" \
8 " [4] sair" )

[ $? -ne 0 ] && sair

case $m in
1) script_1 ;;
2) script_2 ;;
3) script_3 ;;
4) sair ;;
esac
}

sair () {
dialog \
--backtitle \
**** Analisador de recursos e usuários **** \
--title '*** Saindo ***' \
--infobox '\n Obrigado por usar Sistema \n' \
6 70 \
[ $? -ne 0 ] && sair
sleep 5
clear
exit
}

# *****
# ** QUESTÃO PROBLEMA - 01 - script_1 **
# *****

script_1 () {
dialog \
--backtitle \
--stdout \
**** Analisador de recursos e usuários **** \
--title '*** DESEJA SOMAR ***' \
" **** DESEJA SOMAR **** " \
--yesno \
"\nContinuar ?" \
0 0

if [ "$?" = "0" ];
then
# RECEBENDO O PRIMEIRO NUMERO
N1=$( dialog \
--stdout \
--backtitle \
**** Analisador de recursos e usuários **** \
--title '*** Adição (Soma) ***' \
" *** Adição (Soma) *** " \
--inputbox \
"\nDigite o primeiro número: \n" \
0 0 )

if [ -z $N1 ]
then
dialog \
--backtitle \
**** Analisador de recursos e usuários **** \
--title '*** ERRO ***' \
--infobox '\nValor informado inválido - reinicie\' \
0 0
sleep 3
clear
script_1
else
if (echo $N1 | egrep '^[^0-9]' &> /dev/null)
then
dialog \
--backtitle \
**** Analisador de recursos e usuários **** \
--title '*** ERRO ***' \
--infobox '\nValor informado inválido - reinicie\' \
0 0
sleep 3
clear
script_1
else
# RECEBENDO O SEGUNDO NUMERO
N2=$( dialog \
--stdout \
--backtitle \

```

Faculdade de Tecnologia de Ourinhos

```

*** Analisador de recursos e usuários *** \
--title                                     \
" *** Adição (Soma) *** "                 \
--inputbox                                \
"\nDigite o segundo número: \n" \
0 0 )

if [ -z $N2 ]
then
dialog
--backtitle                               \
"*** Analisador de recursos e usuários ***" \
--title ' *** ERRO *** '                 \
--infobox '\nValor informado inválido - reinicie\'
0 0
sleep 3
clear
script_1
else

if (echo $N2 | egrep '[^0-9]' &> /dev/null)
then
dialog
--backtitle                               \
"*** Analisador de recursos e usuários ***" \
--title ' *** ERRO *** '                 \
--infobox '\nValor informado inválido - reinicie\'
0 0
sleep 3
clear
script_1
else

soma=$(echo "scale=2;(($N1 + $N2 ))" | bc)

dialog
--backtitle                               \
"*** Analisador de recursos e usuários ***" \
--title ' *** Resultado *** '             \
--infobox "\n\n A Soma de $N1 + $N2 = $soma" \
6 40
[ $? -ne 0 ] && sair
sleep 5
clear

opcao=$(dialog
--stdout                                  \
--backtitle                               \
"*** Analisador de recursos e usuários ***" \
--title                                     \
"O que deseja fazer agora? "             \
--menu                                     \
"\nEscolha uma opção"                   \
0 0 3                                     \
1 " [1] Continuar"                       \
2 " [2] Menu Principal"                  \
3 " [3] Sair")

[ $? -ne 0 ] && sair

case $opcao in
1) script_1 ;;
2) menu ;;
3) sair ;;
)

esac

fi

sair
fi

sair
fi

sair
fi

clear

}

# *****
# ** QUESTÃO PROBLEMA - 02 - script_2 **
# *****

script_2 () {
clear

dialog
--backtitle                               \
--stdout                                  \
"*** Analisador de recursos e usuários ***" \
--title                                     \
" *** Informações da Máquina *** " \
--yesno                                     \
"\nContinuar ?" \
0 0

if [ "$?" = "0" ];
then

# COLETANDO INFORMAÇÕES

# a) Nome da Máquina,
# b) Data e Hora Atual,
# c) Desde quando a máquina está ativa,
# d) Versão do Kernel,
# e) Quantidade de CPUs/Cores,
# f) Modelo da CPU,
# g) Total de Memória RAM Disponível,
# h) Partições.

```

Faculdade de Tecnologia de Ourinhos

```
# Foi atentado exclusivamente ao solicitado nesta atividade!!!

# Formantando o arquivo de saída com as informações da máquina:
echo "-----" | tee -a arqtmpinfomaquina.txt
echo "    INFORMAÇÕES COLETADAS: " | tee -a arqtmpinfomaquina.txt
echo "-----" | tee -a arqtmpinfomaquina.txt
#a)
hostname | awk '{print"\n    a) Nome da Máquina: \"$1\"'} | tee -a arqtmpinfomaquina.txt
#b)
date | awk '{print"    b) Data e Hora Atual: \"$1,$2,$3,$4,$5\"'} | cut -d: -f1,2,3 | tee -a arqtmpinfomaquina.txt
#c)
uptime | awk '{print"    c) Desde quando a máquina está ativa: Está ativa a \"$3\"'} | cut -d, -f1 | tee -a arqtmpinfomaquina.txt
#d)
uname -mrs | awk '{print"    d) Versão do Kernel: \"$1,$2,$3\"'} | tee -a arqtmpinfomaquina.txt
#e)
CPUs=`lscpu | grep 'CPU(s)' | head -n1 | cut -d" " -f27-35`
Cores=`lscpu | grep 'Core(s)' | cut -d" " -f17-20`
echo "    e) Quantidade de CPUs/Cores: $CPUs/$Cores" | tee -a arqtmpinfomaquina.txt
#f)
modelo=`lscpu | grep 'Model name' | cut -d" " -f24-35`
echo "    f) Modelo da CPU: $modelo" | tee -a arqtmpinfomaquina.txt
#g)
free -h | grep Mem | awk '{print"    g) Total de Memória RAM Disponível: \"$4\"'} | tee -a arqtmpinfomaquina.txt
#h)
echo "    h) Partições:" | tee -a arqtmpinfomaquina.txt
echo " " | tee -a arqtmpinfomaquina.txt
df -h | tee -a arqtmpinfomaquina.txt
echo "-----" | tee -a arqtmpinfomaquina.txt
dialog
--backtitle
--title "*** Analisador de recursos e usuários ***"
--textbox arqtmpinfomaquina.txt
20 70

rm arqtmpinfomaquina.txt
clear

opcao=$(dialog
--stdout
--backtitle
"*** Analisador de recursos e usuários ***"
--title
"O que deseja fazer agora? "
--menu
"\nEscolha uma opção"
0 0 3
1 " [1] Continuar"
2 " [2] Menu Principal"
3 " [3] Sair")

[ $? -ne 0 ] && sair

case $opcao in
1) script_2 ;;
2) menu ;;
3) sair ;;
esac

else
sair
fi
}

# *****
# ** QUESTÃO PROBLEMA - 03 - script_3
# *****

script_3 () {
clear

dialog
--backtitle
--stdout
"*** Analisador de recursos e usuários ***"
--title
" *** Verificar Usuário *** "
--yesno
"\nContinuar ?"
0 0

if [ "$?" = "0" ];
then

US=$(dialog
--stdout
--backtitle
"*** Analisador de recursos e usuários ***"
--title
" *** Verificador de Usuário *** "
--inputbox
"Informe o Usuário: \n\"
0 0
)

if [ -z $US ]
then
#f)

dialog
--backtitle
--title "*** Analisador de recursos e usuários ***"
--title " *** ERRO *** "
--infobox "\nValor informado inválido - reinicie\"
0 0
sleep 3
clear
script_3
else
status=`finger $US | grep Login | cut -d: -f1`
```

Faculdade de Tecnologia de Ourinhos

```

clear
if [ $status = "Login" ]
then
# COLETANDO INFORMAÇÕES
# Crie um script que receba um nome de usuário como parâmetro e exiba as seguintes informações:

# a) UID do usuário,
# b) Nome Completo
# c) Descrição do Usuário,
# d) Total em Uso no /home do usuário,
# e) Informações do último login do usuário.
# f) Opcional: Validar se o usuário existe ou não.
echo "-----" | tee -a arqtmpuser.txt

echo "    Dados do Usuário: " | tee -a arqtmpuser.txt
echo "-----" | tee -a arqtmpuser.txt

#a)
id -u $US | awk '{print"\n    a) UID do usuário: "$1}' | tee -a arqtmpuser.txt
#b)
name=`finger $US | grep Name | cut -d: -f3`
echo "    b) Nome do Usuário: $name" | tee -a arqtmpuser.txt
#c)
Tipo=`id -g $US`
echo "$Tipo"
if [ $Tipo = 0 ]
then
desc="Administrativo"
elif [ $Tipo = 100 -o $Tipo = 1000 ]
then
desc="Comun"
else
desc="do Sistema"
fi
echo "    c) Descrição do Usuário: Usuário $desc" | tee -a arqtmpuser.txt
#d)
dir=`finger $US | grep Directory | cut -d: -f2 | cut -d" " -f1-2`
tot=`df -h $dir | awk '{print $5}' | tail -n1`
echo "    d) Total em Uso no /home do usuário: $tot" | tee -a arqtmpuser.txt
#e)
echo "    e) Últimos logins do usuário:" | tee -a arqtmpuser.txt
echo " " | tee -a arqtmpuser.txt
last -F $US | head -n3 | tee -a arqtmpuser.txt
echo "-----" | tee -a arqtmpuser.txt
dialog
--backtitle \
*** Analisador de recursos e usuários *** \
--title ' *** Usuário Existente *** ' \
--textbox arqtmpuser.txt \
20 0 \
rm arqtmpuser.txt
clear

opcao=$(dialog
--stdout \
--backtitle \
*** Analisador de recursos e usuários ***\
--title \
"O que deseja fazer agora? " \
--menu \
"\nEscolha uma opção" \
0 0 3 \
1 " [1] Continuar" \
2 " [2] Menu Principal" \
3 " [3] Sair") \

[ $? -ne 0 ] && sair

case $opcao in
1) script_3 ;;
2) menu ;;
3) sair ;;
esac

else

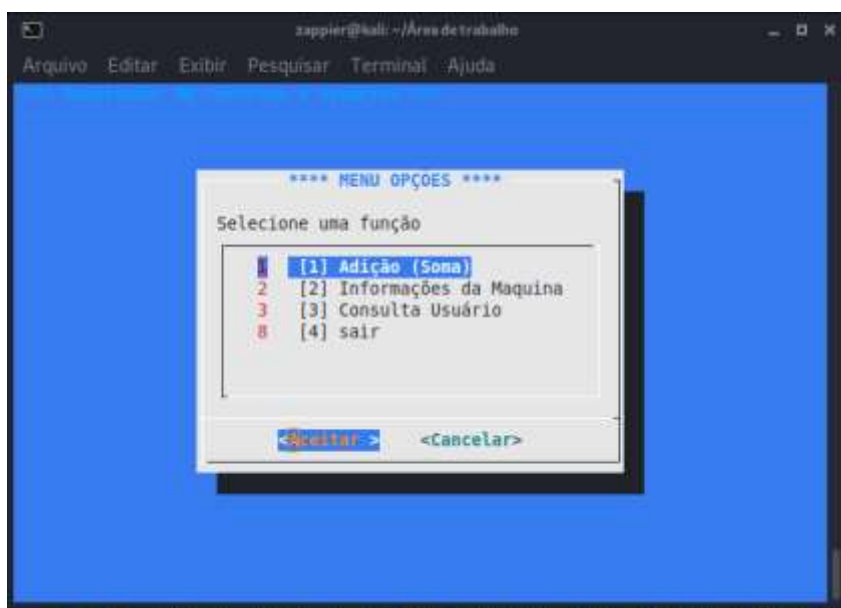
dialog
--backtitle \
*** Analisador de recursos e usuários *** \
--title ' *** ERRO *** ' \
--infobox '\nUsuário inválido - reinicie' \
6 40 \
sleep 3
clear
script_3

fi

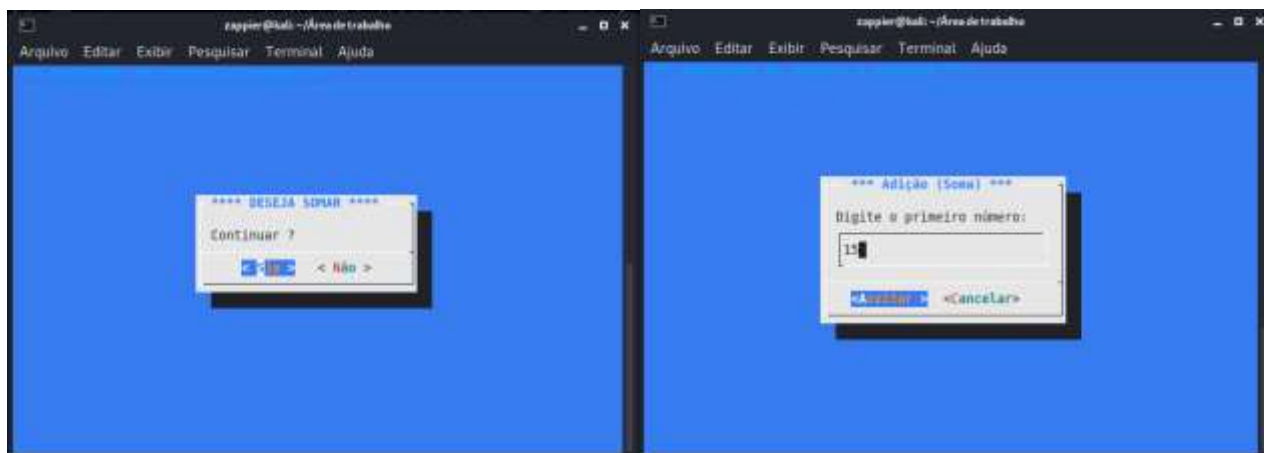
fi
else
sair
fi
}
Menu

```

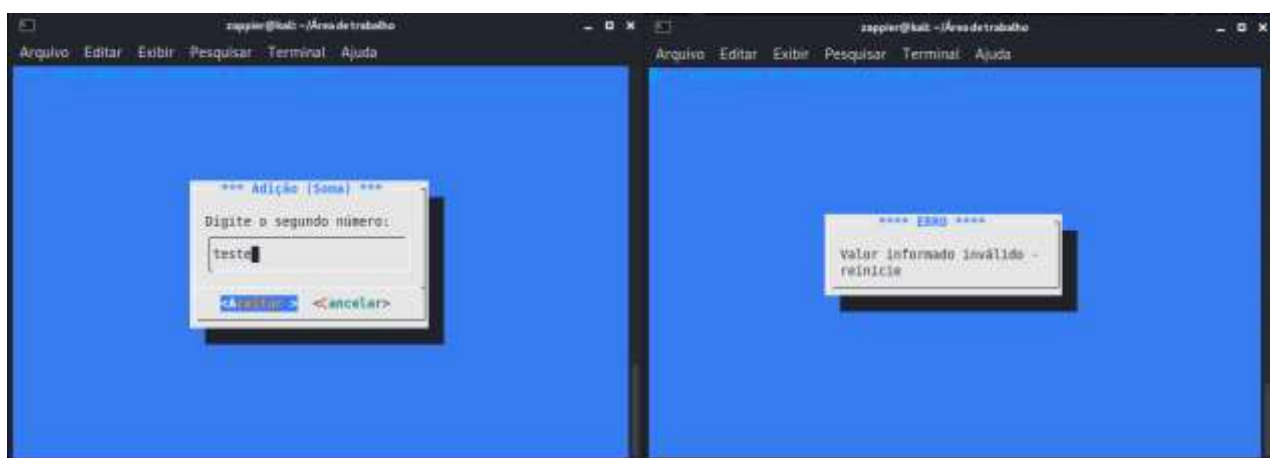
Menu de Opções



Opção [1] Adição (Soma) script_01

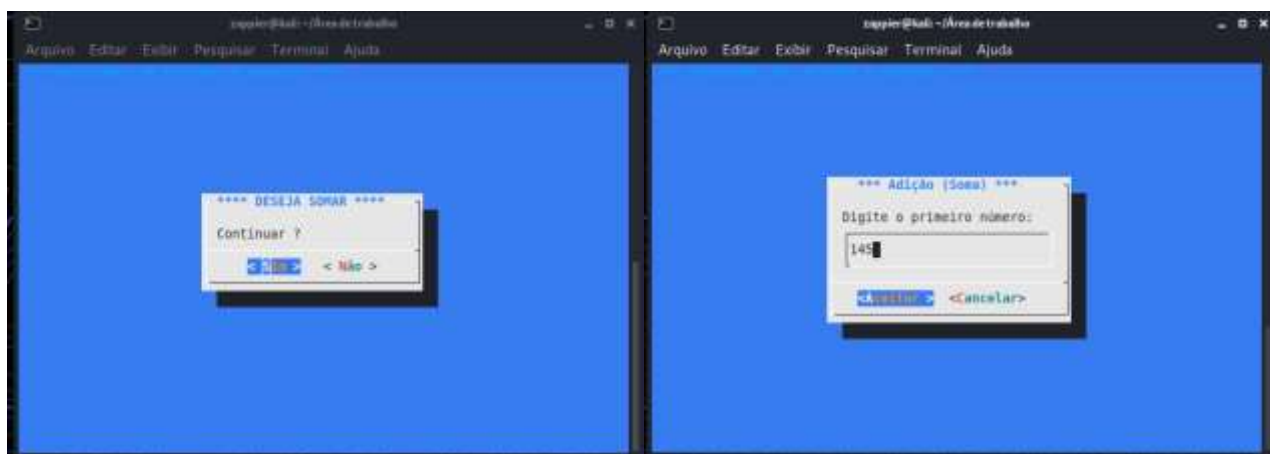


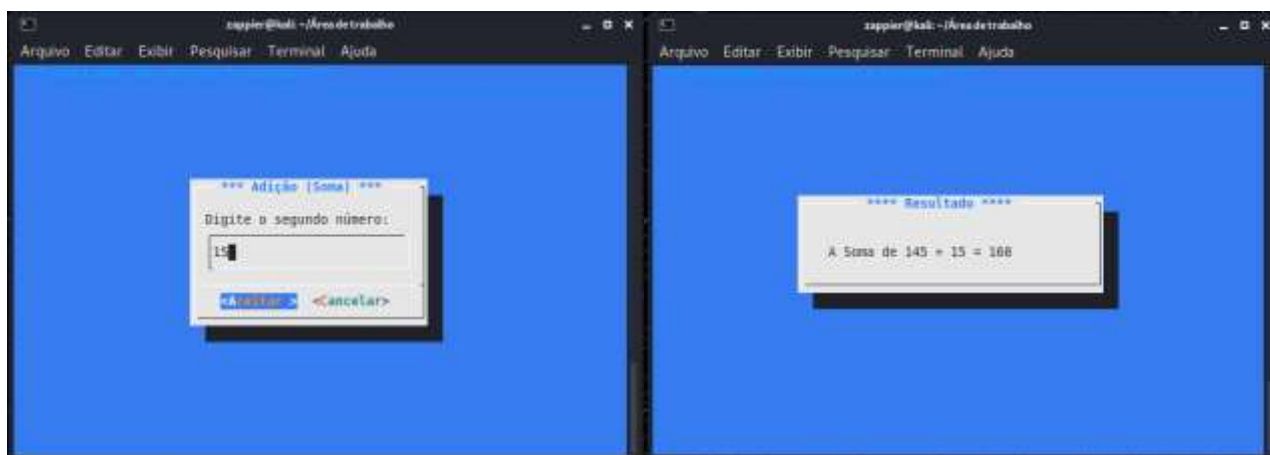
Verificação de campo



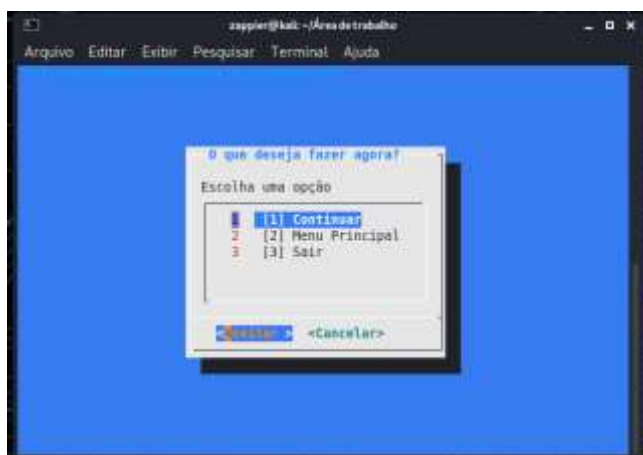
Obs. Nesse caso tanto no campo Número 01, como campo Número 02 o processo em caso de erro é reiniciado.

Reset da operação



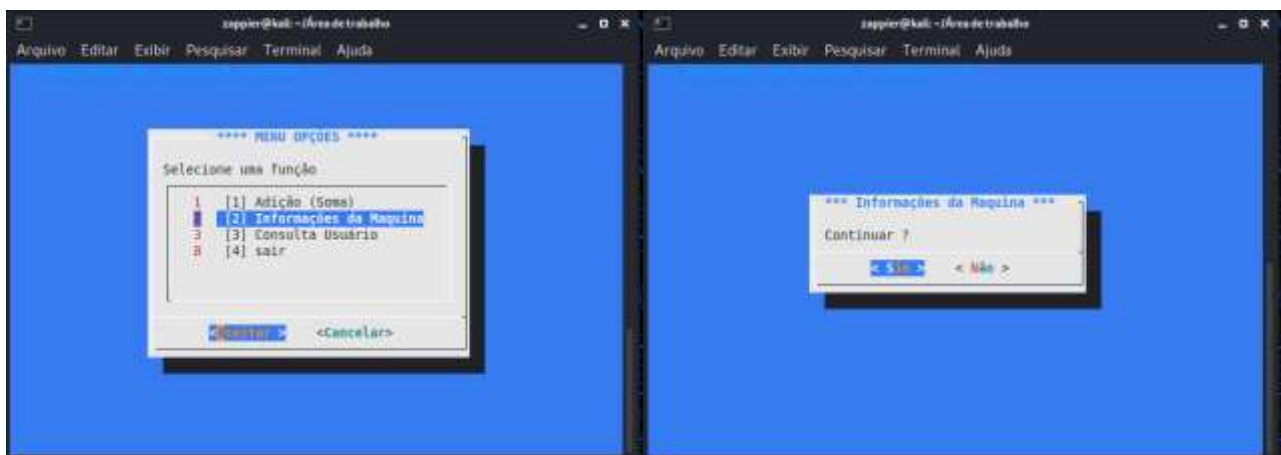


Escolha de opções após amostragem do resultado.



Neste Momento o usuário tem a opção de continuar no script_01 – focado no calculo de soma, sair do sistema ou simplesmente retornar ao menu principal e escolher outra opção.

Opção [2] Informações da maquina
script_02



Obs. Foi inserido campos de confirmações para cobrar do usuário se ele realmente deseja continuar a operação. O objetivo dessa validação é compreender as entradas positivas e negativas, e forçar o usuário a entender o que ele realmente desta por fazer.

Futuramente se pode utilizar esse elo em outros programas onde essa validação será mais do que necessária, como inclusão de condicionais em firewall, compartilhamento de dados em redes e muito mais.

Informações coletadas.

Faculdade de Tecnologia de Ourinhos

```

*** Informações da Máquina ***

INFORMAÇÕES COLETADAS:

a) Nome da Máquina: kali
b) Data e Hora Atual: seg 30 ago 2021 10:04
c) Desde quando a máquina está ativa: Está ativa a 44
d) Versão do Kernel: Linux 5.10.0-kali9-amd64 x86_64
e) Quantidade de CPUs/Cores: 1/1
f) Modelo da CPU: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz
g) Total de Memória RAM Disponível: 2,9Gi
h) Partições:

Sist. Arq.      Tam. Usado Disp.  Use% Montado em
udev           1,9G   0 1,9G   0% /dev
tmpfs          394M  960K  393M   1% /run

*** Informações da Máquina ***

c) Desde quando a máquina está ativa: Está ativa a 44
d) Versão do Kernel: Linux 5.10.0-kali9-amd64 x86_64
e) Quantidade de CPUs/Cores: 1/1
f) Modelo da CPU: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz
g) Total de Memória RAM Disponível: 2,9Gi
h) Partições:

Sist. Arq.      Tam. Usado Disp.  Use% Montado em
udev           1,9G   0 1,9G   0% /dev
tmpfs          394M  960K  393M   1% /run
/dev/sda1       19G   14G  4,0G   79% /
tmpfs           2,0G   0 2,0G   0% /dev/shm
tmpfs           5,0M   0 5,0M   0% /run/lock
tmpfs          394M   0K  394M   1% /run/user/1000
  
```

Escolha de opções após amostragem das informações da máquina.

```

O que deseja fazer agora?

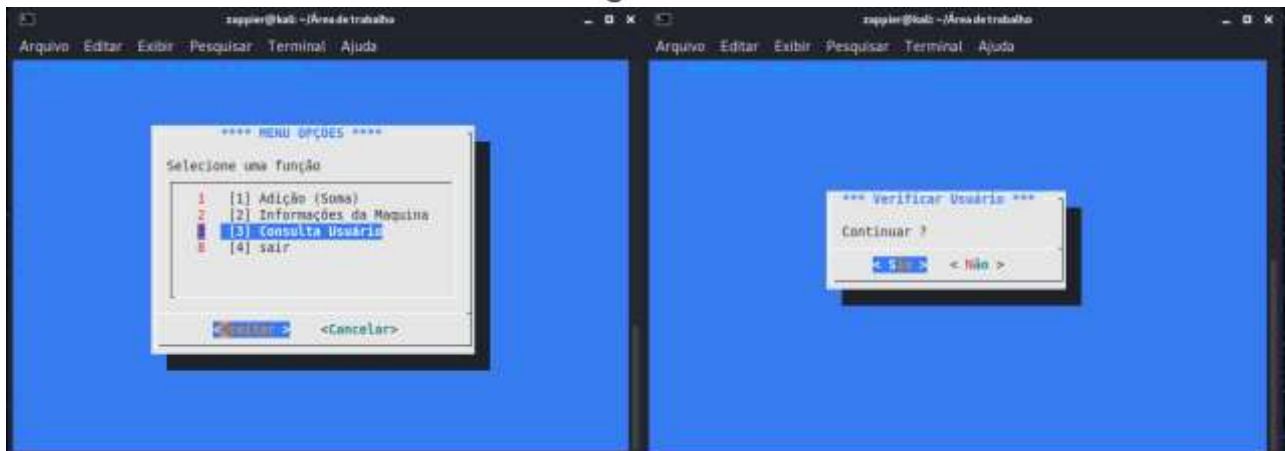
Escolha uma opção

1 [1] Continuar
2 [2] Menu Principal
3 [3] Sair

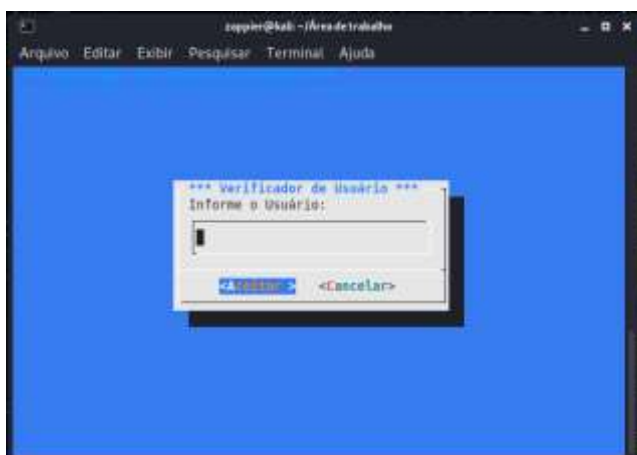
<Enter> <Cancelar>
  
```

Neste Momento o usuário tem a opção de continuar no script_02 – focado no calculo de soma, sair do sistema ou simplesmente retornar ao menu principal e escolher outra opção.

Opção [3] Consulta de Usuários script_03

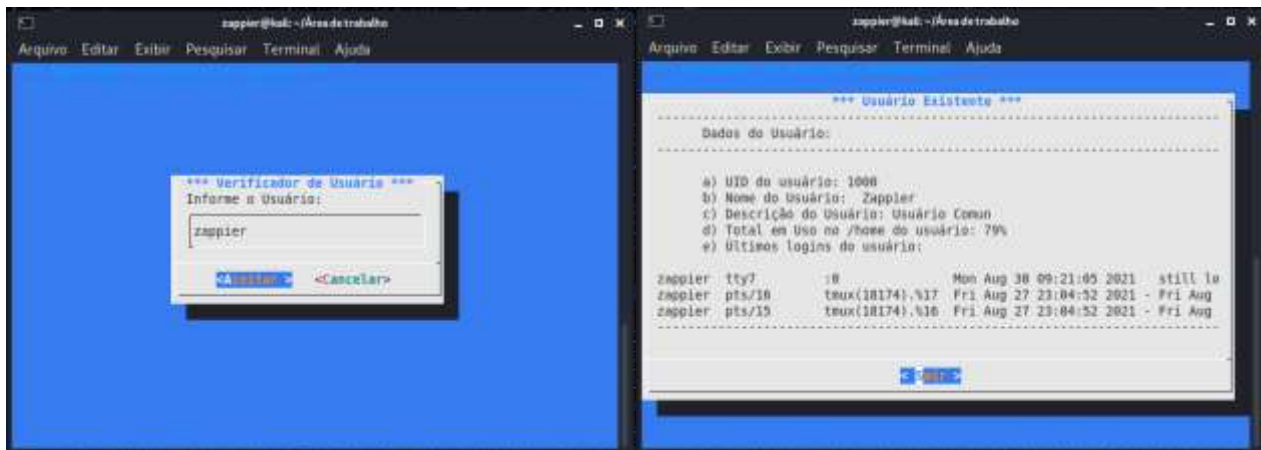
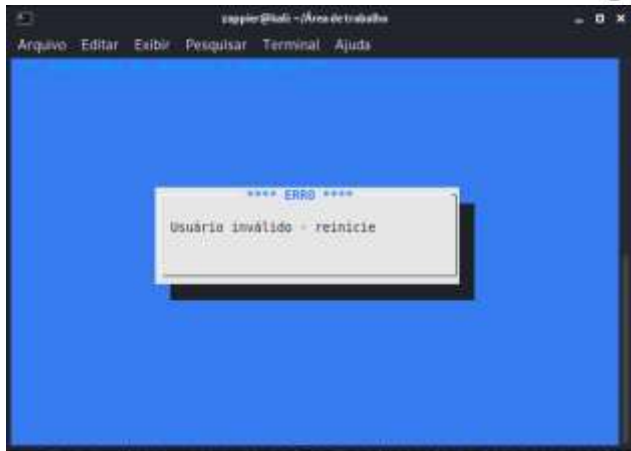


Verificação de usuário e validação de existência.



Neste momento é feita uma verificação na base de dados de usuário se o mesmo de fato existe, mesmo que seja um usuário de sistema. O objetivo é verificar se ele é válido.

Faculdade de Tecnologia de Ourinhos



Obs. A validação do tipo de usuário foi levando-se em consideração o grupo primário do usuário.

Os tipos de usuário do sistema operacional Linux são de três:

- 1) Administrativos (root) - Gid 0
- 2) Usuários comum - Gid 100 (pois existe um grupo padrão onde os usuários comuns podem ser inseridos nele) ou 1000 como nesse caso de teste foi o grupo de usuário criado para o usuário testado.
- 3) Usuários de sistema - Gid (qualquer outro número)

Faculdade de Tecnologia de Ourinhos

```
(zappier@kali) - [~]  
$ id root  
uid=0(root) gid=0(root) grupos=0(root),142(kaboxer)  
  
(zappier@kali) - [~]  
$ id zappier  
uid=1000(zappier) gid=1000(zappier) grupos=1000(zappier),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),109(netdev),118(bluetooth),133(scanner),142(kaboxer)  
  
(zappier@kali) - [~]  
$ id www-data  
uid=33(www-data) gid=33(www-data) grupos=33(www-data)
```

Opção [4] Saída do sistema

