

INF14383L

1º Teste - Treino - Parte 1

1º Teste - Treino - Parte 1

Iniciada terça-feira, 19 de março de 2024 às 09:55

Estado Terminada

Terminada terça-feira, 19 de março de 2024 às 10:20

Tempo gasto 24 minutos 55 segundos

Pergunta 1

Correta

Nota: 1,00

Marcar pergunta

Selecione a opção correta.

- a. O comando **history** mostra um resumo da história do linux.
- b. o comando **!1** repete o último comando.
- c. O comando **!!** repete o último comando. ✓
- d. Nenhuma das anteriores

A sua resposta está correta.

Resposta correta: O comando **!!** repete o último comando.

Pergunta 2

Correta

Nota: 1,00

Marcar pergunta

Indique a opção que melhor descreve a função dos comandos **cp**, **mv**, **mkdir**, e **pwd**.

- a. cp - copia; mv - renomeia; mkdir - cria um novo diretório; pwd - indica o diretório atual ✓
- b. cp - copia; mv - renomeia; mkdir - indica o diretório atual ; pwd - cria um novo diretório
- c. cp - apaga; mv - move; mkdir - cria um novo diretório; pwd - indica o diretório atual
- d. cp - muda o diretório; mv - move; mkdir - indica o diretório atual ; pwd - cria um novo diretório

A sua resposta está correta.

Resposta correta: cp - copia; mv - renomeia; mkdir - cria um novo diretório; pwd - indica o diretório atual

Pergunta 3

Incorreta

Nota: 1,00

▼ Marcar pergunta

Indique o comando ou a combinação de comandos **head** e **tail** que permite mostra as linhas 3, 4, e 5 de um ficheiro de texto.

Exemplo:

```
$ cat linhas.txt
Linha 1
Linha 2
Linha 3
Linha 4
Linha 5
Linha 6
Linha 7

$ < comando | comandos a definir>
Linha 3
Linha 4
Linha 5
```

Resposta: X

Resposta correta: head -n 5 linhas.txt | tail -n 3

Pergunta 4

Incorreta

Nota: 1,00

▼ Marcar pergunta

Indique o comando que adiciona no ficheiro `~/HOME/fcad.log` a seguinte mensagem:

"info: HH:MM"

onde HH e MM são as horas e minutos atuais.

```
$man date

      date - print or set the system date and time

SYNOPSIS

      date [OPTION]... [+FORMAT]
      date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

FORMAT controls the output. Interpreted sequences are:

      %%      a literal %
      %a      locale's abbreviated weekday name (e.g., sun)
      %A      locale's full weekday name (e.g., Sunday)
      %b      locale's abbreviated month name (e.g., Jan)
```

Pergunta 4

Incorreta

Nota: 1,00

Marcar pergunta

Indique o comando que adiciona no ficheiro /\$HOME/fcad.log a seguinte mensagem:

"info: HH:MM"

onde HH e MM são as horas e minutos atuais.

\$man date

date - print or set the system date and time

SYNOPSIS

date [OPTION]... [+FORMAT]

date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

FORMAT controls the output. Interpreted sequences are:

%% a literal %

%a locale's abbreviated weekday name (e.g., Sun)

%A locale's full weekday name (e.g., Sunday)

%b locale's abbreviated month name (e.g., Jan)

%B locale's full month name (e.g., January)

%c locale's date and time (e.g., Thu Mar 3 23:05:25 2005)

%C century; like %Y, except omit last two digits (e.g., 20)

%d day of month (e.g., 01)

%D date; same as %m/%d/%y

%S second (00..60)

%e day of month, space padded; same as %_d

%F full date; like %+4Y-%m-%d

%H hour (00..23)

%M minute (00..59)

%T time; same as %H:%M:%S

(.....)

Resposta: echo "info: \$(date +'%H:%M')" >> \$HOME/fcad.log



Resposta correta: echo "info: \$(date +%H:%M)" >> /\$HOME/fcad.log



1º Teste - Treino - Parte 2

Iniciada terça-feira, 19 de março de 2024 às 10:27

Estado Terminada

Terminada terça-feira, 19 de março de 2024 às 15:35

Tempo gasto 5 horas 8 minutos

Nota Por avaliar

Pergunta 1

Respondida

Nota: 1,00

Marcar pergunta

Crie um script que recebe um e apenas um argumento, e:

- verifica se o numero de argumentos é correcto, se nao for mostra essa indicação ao utilizador e termina
- se o argumento indicar um ficheiro existente mostra o conteúdo (comando **cat**)
- se o argumento indicar um diretório existente list o seu conteúdo (comando **ls**)
- caso contrário, mostra essa indicação ao utilizador e termina

!/bin/bash

```
if [ $# -ne 1 ]; then
```

```
echo Erro
```

```
exit (1)
```

```
fi
```

```
if [ -f $1 ]; then
```

```
cat $1
```

```
exit 0
```

```
fi
```

```
if [ -d $1 ]; then
```

```
ls $1
```

```
exit 0
```

```
fi
```

```
echo Erro
```

```
exit 1
```

```
#!/bin/bash

# Check if exactly one argument is provided
if [ "$#" -ne 1 ]; then
    echo "Usage: $0 <file_or_folder>"
    exit 1
fi

# Check if the argument is a file
if [ -f "$1" ]; then
    cat "$1"
    exit 0
fi

# Check if the argument is a folder
if [ -d "$1" ]; then
    ls "$1"
    exit 0
fi

# If neither a file nor a folder, show an error message
echo "Error: '$1' is neither a file nor a folder."
exit 1
```

Pergunta 2

Respondida

Nota: 1,00

Marcar
pergunta

Crie um script que recebe como argumento o caminho para um ficheiro de utilizadores no seguinte formato:

```
$ cat users_list
arodrigues
malves
pbrito
```

(Um nome de utilizador em cada linha)

Para cada utilizador é criada uma pasta com o nome do utilizador e dentro dessa pasta é criada um ficheiro chamado *user.dat* com a seguinte informação.

```
nome_utilizador nr_caracteres nr_linha
```

Por exemplo, para o utilizador pbrito o ficheiro *user.dat* seria:

```
pbrito 6 3
```

!/bin/bash

```
#EXERCÍCIO 1
#!/bin/bash

# Check if exactly one argument is provided
if [ "$#" -ne 1 ]; then
    echo "Usage: $0 <file_or_folder>"
    exit 1
fi

# Check if the argument is a file
if [ -f "$1" ]; then
    cat "$1"
    exit 0
fi

# Check if the argument is a folder
if [ -d "$1" ]; then
    ls "$1"
    exit 0
fi

# If neither a file nor a folder, show an error message
echo "Error: '$1' is neither a file nor a folder."
exit 1
```

Pergunta 2

Respondida

Nota: 1,00

Marcar pergunta

Crie um script que recebe como argumento o caminho para um ficheiro de utilizadores no seguinte formato:

```
$ cat users_list
arodrigues
malves
pbrito
```

(Um nome de utilizador em cada linha)

Para cada utilizador é criada uma pasta com o nome do utilizador e dentro dessa pasta é criada um ficheiro chamado *user.dat* com a seguinte informação.

```
nome_utilizador nr_caracteres nr_linha
```

Por exemplo, para o utilizador pbrito o ficheiro *user.dat* seria:

```
pbrito 6 3
```

!/bin/bash

```
count = 1
cat $1 | while read line
do
mkdir $line
echo -e "$line \t $(echo $line | wc -c) \t $count" > $line/user.dat
count = ${count+1}
done
```

Pergunta 3

Respondida

Nota: 1,00

Marcar pergunta

Crie um script bash que 1 vez por segundo verifica se existem processos do programa *my_program* em execução. Caso existam e estejam em execução há mais de 60 segundos termina o processo com SIGTERM.

!/bin/bash

```
while true;
```



```
#EXERCÍCIO 2
#!/bin/bash Ficheiros < Arquivos de texto / bestei_treino : 1

# Check if the argument is provided
if [ $# -eq 0 ]; then
    echo "Usage: $0 <file_path>" args.sed
    exit 1 file_or_fold
fi er.sh linhas.txt process_mo
                                nitor.sh

# Check if the file exists
if [ ! -f "$1" ]; then
    echo "File $1 does not exist."
    exit 1
fi

# Read the file line by line and create a folder for each line
count=1
cat "$1" | while read line; do
    echo $line
    if [ ! -z "$line" ]; then
        mkdir -p "$line"
        echo "Created folder: $line"
        echo "$line ${$(echo $line | wc -c)-1} $count" > $line/user.dat
        ((count++))
    fi
done
```

Pergunta 3

Respondida

Nota: 1,00

Marcar
pergunta

Crie um script bash que 1 vez por segundo verifica se existem processos do programa *my_program* em execução. Caso existam e estejam em execução há mais de 60 segundos termina o processo com SIGTERM.

!/bin/bash

```
while true;
do
    if [ ps | grep "my_program" ]; then
        pids=$(ps | grep "my_program" | awk '{print $1}')

        for pid in $pids; do
            runtime=$(ps -o etime= $pid | awk -F'.' '($1 * 60) + $2')
            if [ $runtime -gt 60 ]; then
                kill -15 $pid
            fi
        done
    fi
    sleep 1
done
```

Terminar revisão

Os seus dados de acesso são iguais aos que lhe dão acesso ao SIIUE (nome de utilizador e palavra-passe válida).

Para experimentar a utilização desta plataforma do ponto de vista do aluno e obter algumas informações, aceda à área "Laboratório Moodle".

Informação

[Universidade de Évora](#)

[Serviços de Informática](#)

Contacte-nos

Telefone : + 351 266 740 800

E-mail : moodle@uevora.pt

```
#EXERCÍCIO 3
#!/bin/bash

while true; do
    # Get the process ID of my_process
    spids=$(ps -eo pid,cmd | grep "my_process" | cut -d' ' -f1)

    for pid in $spids; do
        if [ -n "$pid" ]; then
            # Get the execution time of the process
            etime=$(ps -o etime= -p $pid | tr -d ' ')
            # Extract hours, minutes, and seconds from the execution time
            hours=$(echo $etime | cut -d: -f1)
            minutes=$(echo $etime | cut -d: -f2)
            seconds=$(echo $etime | cut -d: -f3)
            # Convert execution time to seconds
            total_seconds=$((hours*3600 + minutes*60 + seconds))
            # If execution time is more than 60 seconds, send SIGTERM
            if [ $total_seconds -gt 60 ]; then
                echo "Process my_process with PID $pid has been running for more than 60 seconds. Killing with SIGTERM..."
                kill -15 $pid
            fi
        done
        sleep 1
    done
done
```

INF14383L

Exercícios Treino - Bash

Exercícios Treino - Bash

Iniciada segunda-feira, 1 de abril de 2024 às 21:26

Estado Terminada

Terminada segunda-feira, 1 de abril de 2024 às 22:26

Tempo gasto 1 hora

Nota Por avaliar

Pergunta 1

Incorreta

Nota: 0,00 em

1,00

▼ Marcar pergunta

Qual dos scripts abaixo lê todas as linhas do arquivo *linhas.txt* e imprime o número da linha seguido do conteúdo?

`./script1.sh`

`#!/bin/bash`

`count=1`

```
cat linhas.txt | while read line; do
    echo "$count $line"
    ((count++))
done
```

`./script2.sh`

`#!/bin/bash`

`count=1`

```
while read line; do
    echo "$count $line"
    count=$((count + 1))
done < "linhas.txt"
```

- a. Ambos os scripts
- b. Apenas o script2.sh X
- c. Apenas o script1.sh
- d. Nenhum dos scripts

A sua resposta está incorreta.

Resposta correta: Ambos os scripts



Pergunta 2

Correta

Nota: 1,00 em
1,00

Marcar
pergunta

Qual a diferença entre os comandos **kill** e **killall**?

- a. Não há diferenças, ambos "matam" um processo pelo seu PID
- b. **kill** "mata" um processo pelo seu PID, enquanto **killall** "mata" um processo pelo seu nome ✓
- c. **killall** "mata" um processo pelo seu PID, enquanto **kill** "mata" um processo pelo seu nome
- d. Nenhuma das anteriores

A sua resposta está correta.

Resposta correta:

kill "mata" um processo pelo seu PID, enquanto **killall** "mata" um processo pelo seu nome

Pergunta 3

Parcialmente
correta

Nota: 0,67 em
1,00

Marcar
pergunta

Qual o comando que pode ser usado para verificar se um arquivo existe?

Escolha todas as alternativas corretas.

- a. **test** -d arquivo.txt
- b. **test** -e arquivo.txt ✓
- c. **test** arquivo.txt
- d. **test** -f arquivo.txt ✓

A sua resposta está parcialmente correta.

Selecionou 2 respostas corretas.

Respostas corretas:

test -f arquivo.txt

,

test arquivo.txt

,

test -e arquivo.txt

Pergunta 4

Correta

Qual dos comandos abaixo é usado para dar permissões de execução ao arquivo script.sh?

Pergunta 4

Correta

Nota: 1,00 em
1,00

▼ Marcar
pergunta

Qual dos comandos abaixo é usado para dar permissões de execução ao arquivo script.sh?

Escolha todas as alternativas corretas.

- a. chmod 000 script.sh
- b. chmod +x script.sh ✓
- c. chmod 777 script.sh ✓
- d. chmod -x script.sh

A sua resposta está correta.

Respostas corretas:

chmod +x script.sh

✓
chmod 777 script.sh

Pergunta 5

Correta

Nota: 1,00 em
1,00

▼ Marcar
pergunta

Qual a afirmação correcta sobre os comandos **fg** e **bg**?

- a. Nenhuma das anteriores
- b. **fg** coloca um processo em foreground, enquanto **bg** coloca um processo em background ✓
- c. Não há diferença, ambos gerem processos de forma equivalente.
- d. **fg** coloca um processo em background, enquanto **bg** coloca um processo em foreground

A sua resposta está correta.

Resposta correta:

fg coloca um processo em foreground, enquanto **bg** coloca um processo em background

Pergunta 6

Correta

Como se pode colocar um processo em suspensão?

Pergunta 6

Correta

Nota: 1,00 em
1,00

▼ Marcar
pergunta

Como se pode colocar um processo em suspensão?

Escolha todas as alternativas corretas.

- a. Ctrl + z ✓
- b. kill -TERM PID
- c. Nenhuma das anteriores
- d. kill -STOP PID ✓

A sua resposta está correta.

Respostas corretas: Ctrl + z,
kill -STOP PID

Pergunta 7

Correta

Nota: 1,00 em
1,00

▼ Marcar
pergunta

O que faz o comando kill -9 PID?

- a. Termina o processo ✓
- b. Coloca o processo em background
- c. Nenhum dos anteriores
- d. Coloca o processo em foreground

A sua resposta está correta.

Resposta correta:
Termina o processo

Pergunta 8

Correta

Nota: 1,00 em
1,00

▼ Marcar
pergunta

Qual o resultado do script?

```
a=1
if [ $a -eq 1 ]; then
    echo "a == 1"
else
    echo "a != 1"
fi
```

Pergunta 8

Correta

Nota: 1,00 em

1,00

▼ Marcar pergunta

Qual o resultado do script?

```
a=1
if [ $a -eq 1 ]; then
    echo "a == 1"
else
    echo "a != 1"
fi
```

- a. Nenhuma das anteriores
- b. a == 1 /n a != 1
- c. a != 1
- d. a == 1 ✓

A sua resposta está correta.

Resposta correta:

a == 1

Pergunta 9

Correta

Nota: 1,00 em

1,00

▼ Marcar pergunta

Qual o resultado do script?

```
c=1
while [ $c -le 5 ]; do
    echo -n "$c "
    c=$[ $c + 1 ]
done
```

- a. Nenhuma das anteriores
- b. 1 2 3 4 5 ✓
- c. 12345
- d. 1 2 3 4

A sua resposta está correta.

Resposta correta:

1 2 3 4 5



Pergunta 10

Correta

Nota: 1,00 em
1,00

▼ Marcar
pergunta

Qual o output do script?

```
a=1
if [ $a -eq 2 ]; then
    echo "Then"
else
    echo "Else"
fi
```

Resposta: Then



Resposta correta: Then

Pergunta 11

Respondida

Nota: 1,00

▼ Marcar
pergunta

Crie um script que leia um arquivo chamado numeros.txt e imprima a soma de todos os numeros do arquivo.

```
#!/bin/bash
if [ -e "numeros.txt" ]; then
    soma=0
    while read -r numero; do
        soma=$((soma + numero))
    done < "numeros.txt"
fi
echo "A soma de todos os números do arquivo é: $soma"
```

Pergunta 12

Respondida

Nota: 1,00

Marcar pergunta

Crie um script que leia um arquivo chamado numeros.txt e imprima a soma de todos os números pares do arquivo.

```
#!/bin/bash
if [ -e "numeros.txt" ]; then
    soma=0
    while read -r numero; do
        if (( numero % 2 == 0 )); then
            soma=$((soma + numero))
        fi
    done < "numeros.txt"
fi
echo "A soma dos números pares é: $soma"
```

Pergunta 13

Respondida

Nota: 1,00

Marcar pergunta

Crie um script que procura por arquivos com extensão .txt em um diretório e imprime o nome dos arquivos encontrados.

```
#!/bin/bash
diretorio="$PATH"
if [ -d "$diretorio" ]; then
    for file in *.txt;
    do
        if [ -f $file ]; then
            echo $file
        fi
    done
fi
```

Pergunta 14

Respondida

Nota: 1,00

Marcar pergunta

Crie um script que produz um arquivo chamado numeros.txt com 10 números aleatórios entre 1 e 100.

Dica: use a variável \$RANDOM

do manual:

RANDOM Each time this parameter is referenced, it expands to a random integer between 0 and 32767. Assigning a value to RANDOM initializes (seeds) the sequence of random numbers. If RANDOM is unset, it loses its special properties, even if it is subsequently reset.

```
#!/bin/bash
echo "" > numeros.txt
for ((i = 0; i < 10; i++));
do
    numero=$((RANDOM % 100 + 1))
    echo "$numero" >> "numero.txt"
done
```

Pergunta 15

Respondida

Nota: 1,00

Marcar pergunta

Crie um script que leia todos os ficheiros .csv num diretório e gere um ficheiro chamado all3_6.csv com as colunas 3 e 6 de todos os ficheiros .csv.

O script recebe como argumento o diretório onde estão os ficheiros .csv. Se o diretório não for passado como argumento, o script deve usar o diretório atual.

Exemplo de uso: ./script.sh /home/user/csv_files

Dica: use o comando cut

```
#!/bin/bash
if [ $# -eq 0 ]; then
```

Pergunta 15

Respondida

Nota: 1,00

Marcar
pergunta

Crie um script que leia todos os ficheiros .csv num diretório e gere um ficheiro chamado all3_6.csv com as colunas 3 e 6 de todos os ficheiros .csv.

O script recebe como argumento o diretório onde estão os ficheiros .csv. Se o diretório não for passado como argumento, o script deve usar o diretório atual.

Exemplo de uso: ./script.sh /home/user/csv_files

Dica: use o comando cut

```
#!/bin/bash
if [ $# -eq 0 ]; then
    diretorio=."
else
    diretorio="$1"
fi
echo "" > "all3_6.csv"
for file in *.csv; do
    cut -d "," -f 3,6 "$file" >> "all3_6.csv"
done
```

Terminar revisão