

# Trabalho de Sistemas Operativos

Mario Junior Manhente Portilho ([a27989@alunos.ipca.pt](mailto:a27989@alunos.ipca.pt))

Tiago Daniel Silva Vale ([a27675@alunos.ipca.pt](mailto:a27675@alunos.ipca.pt))

Pedro Alvaro Carvalho Duarte ([a27990@alunos.ipca.pt](mailto:a27990@alunos.ipca.pt))

Prof. Fernando Gomes



# Índice

<b>Resumo.....</b>	<b>3</b>
<b>Parte 1.....</b>	<b>3</b>
<b>Parte 2.....</b>	<b>5</b>
<b>Parte 3.....</b>	<b>6</b>
Figura 1 – Gerenciamento de partições.....	9
Figura 2 – Iniciação de uma partição.....	10
Figura 3 – Criação de um grupo de volumes.....	10
Figura 4 – Criar volumes lógicos.....	10
Figura 5 – Exibir informações sobre todos os grupos de volumes disponíveis no sistema.....	10
Figura 6 – Criação de sistemas de ficheiros em uma partição.....	11
Figura 7 – Criação de sistemas de ficheiros em uma partição.....	12
Figura 8 – Criação e montagem para sistema de ficheiros.....	13
Figura 9 – Edição de ficheiro.....	13
Figura 10 – Criar ficheiro com nº de alunos.....	14
Figura 11 – Permissão.....	14
Figura 12 – Listagem de permissão.....	14
Figura 13 – Mostra o uso de espaço em disco dos sistemas de ficheiros montados em diretórios.....	14
Figura 14 – Info do fs.img.....	15
Figura 15 – Comando fsstat para info do sistema.....	16
Figura 16 – Fim da informação do sistema (restante continuação de figura anterior).....	17
Figura 17 – Listar inodes dos ficheiros.....	17
Figura 17 – Apresentação do conteúdo de “ipca.txt”.....	17
<b>Conclusão.....</b>	<b>17</b>
<b>Bibliografia.....</b>	<b>18</b>

# Resumo

Este relatório foi elaborado no âmbito da disciplina de Sistemas Operativos lecionada pelo docente Fernando Gomes tendo como objetivo a aplicação prática dos conceitos de gestão de processo e ficheiros, colimando toda a matéria lecionada ao longo do semestre.

**Parte 1** - Mário Portilho

**Parte 2** - Pedro Duarte

**Parte 3** - Tiago Vale

## Parte 1

A parte um do trabalho consistia na implementação de um conjunto de comandos para manipulação de ficheiros através de chamadas ao sistema (*system calls*). Chamadas ao sistema e o mecanismo pelo qual o aplicativo solicita serviços do kernel ao sistema operativo podendo estas ser de hardware como por exemplo, acessar a um disco externo, ou de software como por exemplo, criar, editar e guardar um ficheiro, podemos consultalas em: <https://linux.die.net/man/>.

O conjunto de programas pedido foram:

- **mostra** ficheiro - Este comando deve apresentar no ecrã todo o conteúdo do ficheiro indicado como parâmetro. Caso o ficheiro não exista, o comando deve avisar o utilizador que o ficheiro não existe;
- **copia** ficheiro – Este comando deve criar um novo ficheiro, cujo nome é “*ficheiro.copia*”, cujo conteúdo é uma cópia de (todo) o conteúdo do ficheiro passado como parâmetro no comando, com o nome ficheiro. Caso o ficheiro não exista, deve ser apresentado um aviso ao utilizador;

- **acrescenta** origem destino – Este comando deve acrescentar (todo) o conteúdo da “origem” no final do “destino”. Caso algum dos ficheiros não exista, deve ser apresentado um aviso ao utilizador;
- **conta** ficheiro – Este comando deve contar o número de linhas existentes num ficheiro. Se o ficheiro não existir, deverá ser indicado ao utilizador uma mensagem de erro;
- **apaga** ficheiro – Este comando deve apagar o ficheiro com o nome indicado. No caso de o ficheiro indicado não existir, e apenas, deve ser apresentado um aviso ao utilizador;
- **conta** ficheiro – Este comando deve contar o número de linhas existentes num ficheiro. Se o ficheiro não existir, deverá ser indicado ao utilizador uma mensagem de erro;
- **apaga** ficheiro – Este comando deve apagar o ficheiro com o nome indicado. No caso de o ficheiro indicado não existir, e apenas, deve ser apresentado um aviso ao utilizador;
- **informa** ficheiro – Este comando apresenta apenas a informação do sistema de ficheiros em relação ao ficheiro indicado, tipo de ficheiro (normal, directoria, link, etc.), i-node, utilizador dono em formato textual e datas de criação, leitura e modificação em formato textual;
- **lista** [directoria] – Este comando deve apresentar uma lista de todas as pastas e ficheiros existentes na directoria indicada ou na directoria atual se não especificada. Adicionalmente, deve distinguir ficheiros simples de diretorias através de uma indicação textual.

Algumas das chamadas ao sistema mais utilizadas na realização desta tarefa foram:

- **write:** `ssize_t write(int fildes, const void *buf, size_t nbyte);`  
Esta chamada é muito usada para a escrita de mensagens (*strings* / *char*) para qualquer descritor de ficheiros (ficheiro.txt, CLI, etc), foi utilizada em programas como mostra, acrescenta, etc, etc.
- **open:** `int open (const char* Path, int flags);`  
Esta chamada é muito utilizada para abrir ficheiros, foi utilizada em programas como mostra, acrescenta, etc etc.
- **read:** `size_t read (int fd, void* buf, size_t cnt);`  
Esta chamada é muito utilizada para ler bytes do descritor de ficheiros, foi utilizada em programas como conta, copia.

## Parte 2

A parte dois do trabalho consistia na implementação de um interpretador de linha de comandos, que servisse de substituto do original (Bash shell) e cuja função deste seria ler os caracteres passados na consola e executar esses mesmos como um comando e argumentos do sistema. Um dos requisitos do terminal era que este mostrasse o símbolo “%” como indicação de que está pronto para ler um novo comando do utilizador.

A leitura dos comandos, é realizada na seguinte ordem:

O `fgets` lê o comando que o usuário introduziu, verifica se é o comando “*termina*”, se verificado termina o programa, se não, envia o comando para a função “*execute\_command*”.

Já dentro da função, cria um processo com o `fork()` e verifica se correu bem. Se o “*pid*” for 0 refere-se ao processo filho, dentro deste o

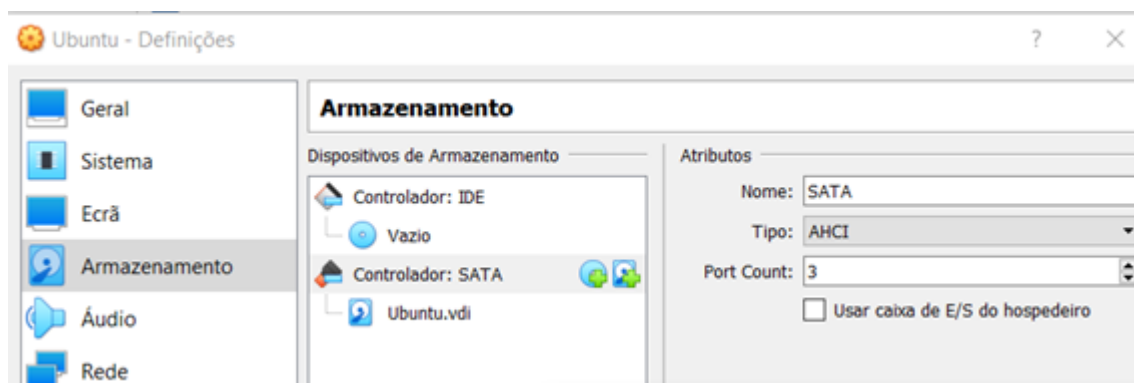
comando é separado em tokens e colocado de novo na variável "args". Após estes passos o "execvp" executa o comando no processo filho. Enquanto isto acontece o processo pai espera pelo filho e quando este acaba verifica se o comando funcionou e mostra o texto "O comando x terminou com o código y.", caso contrário diz que o comando falhou.

## Parte 3

### a) Adicionar um disco novo e criar uma partição:

Objetivo 1: Adicionar um novo disco ao servidor virtual.

Passos: No servidor virtual "Virtual Box" selecionamos uma máquina, neste caso, a máquina definida com o nome "Ubuntu". Deste modo, após o seu selecionamento, clicamos em configurações e logo de seguida armazenamento, em "Controlador: Sata" adicionamos um novo disco virtual (Ubuntu\_A2.vdi), alocado dinamicamente com um tamanho virtual de 10,00 GB. Em síntese, iniciamos a máquina.



? X

← Criar Disco Rígido Virtual

Localização do ficheiro

C:\Users\ASUS\VirtualBox VMs\Ubuntu\Ubuntu\_A2\vd

📁

File size

10,00 GB

4,00 MB
2,00 TB

Hard disk file type

☒ **VDI (VirtualBox Disk Image)**  
☐ **VHD (Virtual Hard Disk)**  
☐ **VMDK (Virtual Machine Disk)**  
☐ HDD (Parallels Hard Disk)  
☐ QCOW (QEMU Copy-On-Write)  
☐ QED (QEMU enhanced disk)

Armazenamento no disco rígido físico

☒ Alocado dinamicamente  
☐ Tamanho fixo  
☐ Split into files of less than 2GB

Modo Assistido

Criar

Cancelar

Ubuntu - Definições

? X

Geral

Sistema

Ecrã

Armazenamento

Áudio

Rede

Portas Série

USB

Pastas Partilhadas

Interface do Utilizador

**Armazenamento**

Dispositivos de Armazenamento

Controlador: IDE

Vazio

Controlador: SATA

Ubuntu.vdi

Ubuntu\_A2.vdi

Atributos

Disco rígido: Porta SATA 1

↻

☐ Unidade de Estado Sólido  
☐ Hot-pluggable

Informação

Tipo (Formato): Normal (VDI)

Tamanho virtual: 10,00 GB

Tamanho atual: 2,00 MB

Detalhes: Dynamically allocated storage

Localização: C:\Users\ASUS\VirtualBox V...

Ligado a: --

Encriptado com chave: --

Objetivo 2: Criar uma partição.

Passos: Na máquina virtual “Ubuntu”, abrimos o terminal e executamos os comandos respetivos a criação de uma partição.

Comandos:

- 1 - `sudo fdisk -l`
- 2 - `sudo fdisk /dev/sdb;`

Extra comandos:

- 1 - digitamos ``m`` para ajuda;
- 2 - digitamos ``n`` para adicionar uma nova partição;
- 3 - digitamos ``p`` para primária;
- 4 - digitamos ``1`` para número da partição;
- 5 - digitamos ``enter`` para confirmação do valor padrão do primeiro setor;
- 6 - digitamos ``enter`` para confirmação do valor padrão do último setor;
- 7 - pressionamos ``w`` para escrever a tabela de partições no disco e sair do ``fdisk``;

```
Disco /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 setores
Disk model: VBOX HARDDISK
Unidades: setor de 1 * 512 = 512 bytes
Tamanho de setor (lógico/físico): 512 bytes / 512 bytes
Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes
```

```
tiago@tiago-VirtualBox:~$ sudo fdisk /dev/sdb
[sudo] senha para tiago:

Bem-vindo ao fdisk (util-linux 2.37.2).
As alterações permanecerão apenas na memória, até que você decida gravá-las.
Tenha cuidado antes de usar o comando de gravação.

A unidade não contém uma tabela de partição conhecida.
Criado um novo rótulo de disco DOS com o identificador de disco 0xa52fd280.

Comando (m para ajuda): m
```



```
Salvar & sair
w  grava a tabela no disco e sai
q  sai sem salvar as alterações

Cria um novo rótulo
g  cria uma nova tabela de partição GPT vazia
G  cria uma nova tabela de partição SGI (IRIX) vazia
o  cria uma nova tabela de partição DOS vazia
s  cria uma nova tabela de partição Sun vazia

Comando (m para ajuda): n
Tipo da partição
  p  primária (0 primary, 0 extended, 4 free)
  e  estendida (recipiente para partições lógicas)
Selecione (padrão p): p
Número da partição (1-4, padrão 1): 1
Primeiro setor (2048-20971519, padrão 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-20971519, padrão 20971519):

Criada uma nova partição 1 do tipo "Linux" e de tamanho 10 GiB.

Comando (m para ajuda): w
```

Figura 1 – Gerenciamento de partições

### Solução:

O disco foi adicionado com sucesso ao servidor virtual e uma partição primária foi criada para posterior uso na criação de um sistema de ficheiros.

### **b) Criar um volume físico e adicionar volumes lógicos:**

Objetivo: Preparar o disco recém-criado para o uso de LVM (Logical Volume Manager).

Passos: No terminal, após a criação de uma partição, criamos um volume físico, criamos um grupo de volumes, criamos volumes lógicos de 5GB cada um.

#### Comandos:

- 1 - sudo pvcreate /dev/sdb1
- 2 - sudo vgcreate vg1 /dev/sdb1
- 3 - sudo lvcreate -L 5G -n lv1 vg1
- 4 - sudo lvcreate -l 100%FREE -n lv2 vg1

```
tiago@tiago-VirtualBox:~$ sudo pvcreate /dev/sdb1
Physical volume "/dev/sdb1" successfully created.
```

Figura 2 – Iniciação de uma partição

```
tiago@tiago-VirtualBox:~$ sudo vgcreate vg1 /dev/sdb1
Volume group "vg1" successfully created
```

Figura 3 – Criação de um grupo de volumes

```
tiago@tiago-VirtualBox:~$ sudo lvcreate -L 5G -n lv1 vg1
Logical volume "lv1" created.
tiago@tiago-VirtualBox:~$ sudo lvcreate -l 100%FREE -n lv2 vg1
Logical volume "lv2" created.
```

Figura 4 – Criar volumes lógicos

```
tiago@tiago-VirtualBox:~$ sudo vgdisplay vg1
--- Volume group ---
VG Name                vg1
System ID
Format                 lvm2
Metadata Areas         1
Metadata Sequence No   17
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 2
Open LV                 0
Max PV                  0
Cur PV                 1
Act PV                  1
VG Size                 <10,00 GiB
PE Size                 4,00 MiB
Total PE                2559
Alloc PE / Size         2559 / <10,00 GiB
Free PE / Size           0 / 0
VG UUID                 URh0LR-prQ1-nq5I-Ioou-UHvj-3XH4-nT0NaX
```

Figura 5 – Exibir informações sobre todos os grupos de volumes disponíveis no sistema

### Solução:

O disco foi preparado como um volume físico e agrupado em um grupo de volumes chamado vg1. Dois volumes lógicos foram criados dentro deste grupo, um com 5GB (lv1) e outro usando todo o espaço livre disponível (lv2).

### **c) Criar sistemas de ficheiros em volumes lógicos:**

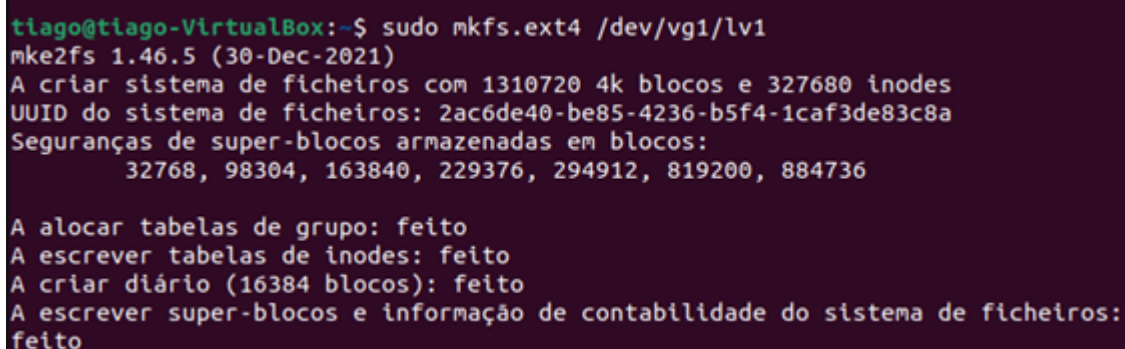
Objetivo: Criar sistemas de arquivos nos volumes lógicos para possibilitar o armazenamento de dados.

Passos: No terminal, criamos um sistema de ficheiros ext4 em `lv1`, de seguida, criamos um sistema de ficheiros ext3 em `lv2`.

### Comandos:

1 - sudo mkfs.ext4 /dev/vg1/lv1

2 - sudo mkfs.ext3 /dev/vg1/lv2



```
tiago@tiago-VirtualBox:~$ sudo mkfs.ext4 /dev/vg1/lv1
mke2fs 1.46.5 (30-Dec-2021)
A criar sistema de ficheiros com 1310720 4k blocos e 327680 inodes
UUID do sistema de ficheiros: 2ac6de40-be85-4236-b5f4-1caf3de83c8a
Seguranças de super-blocos armazenadas em blocos:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

A alocar tabelas de grupo: feito
A escrever tabelas de inodes: feito
A criar diário (16384 blocos): feito
A escrever super-blocos e informação de contabilidade do sistema de ficheiros:
feito
```

Figura 6 – Criação de sistemas de ficheiros em uma partição

```

tiago@tiago-VirtualBox:~$ sudo mkfs.ext3 /dev/vg1/lv2
mke2fs 1.46.5 (30-Dec-2021)
A criar sistema de ficheiros com 1309696 4k blocos e 327680 inodes
UUID do sistema de ficheiros: 4a5d9a74-d9c7-441c-a953-f2a57987f2b3
Seguranças de super-blocos armazenadas em blocos:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

A alocar tabelas de grupo: feito
A escrever tabelas de inodes: feito
A criar diário (16384 blocos): feito
A escrever super-blocos e informação de contabilidade do sistema de ficheiros:
feito

```

Figura 7 – Criação de sistemas de ficheiros em uma partição

### Solução:

Foram criados sistemas de ficheiros ext4 e ext3 nos volumes lógicos lv1 e lv2, respetivamente.

### **d) Montar sistemas de ficheiros e torná-los persistentes:**

Objetivo: Montar os sistemas de arquivos criados nos diretórios desejados e garantir que a montagem seja persistente.

Passo: No terminal, executamos comandos de modo, a criar diretórios de montagem, bem como montar os sistemas de ficheiros, e sucessivamente adicionamos as entradas ao `fstab` para a sua respetiva persistência e guardamos as alterações efetuadas.

### Comandos:

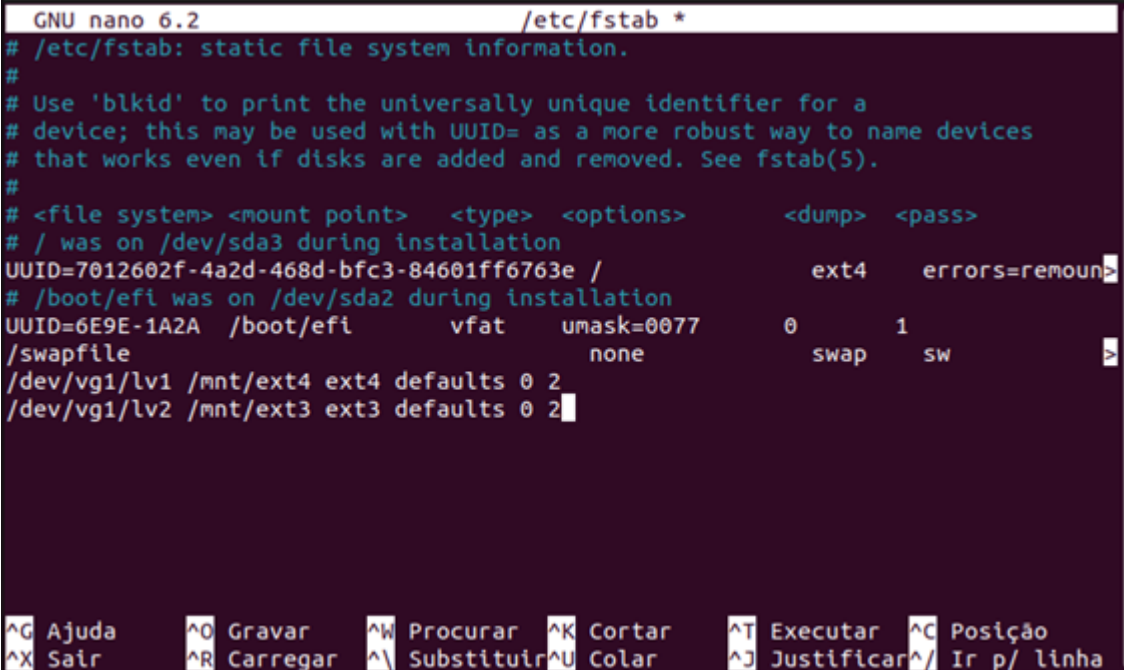
- 1 - sudo mkdir /mnt/ext4
- 2 - sudo mkdir /mnt/ext3
- 3 - sudo mount /dev/vg1/lv1 /mnt/ext4
- 4 - sudo mount /dev/vg1/lv2 /mnt/ext3

```

tiago@tiago-VirtualBox:~$ sudo mkdir -p /mnt/ext4
tiago@tiago-VirtualBox:~$ sudo mkdir -p /mnt/ext3
tiago@tiago-VirtualBox:~$ sudo mount /dev/vg1/lv1 /mnt/ext4
tiago@tiago-VirtualBox:~$ sudo mount /dev/vg1/lv2 /mnt/ext3
tiago@tiago-VirtualBox:~$ sudo nano /etc/fstab

```

Figura 8 – Criação e montagem para sistema de ficheiros



```

GNU nano 6.2 /etc/fstab *
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda3 during installation
UUID=7012602f-4a2d-468d-bfc3-84601ff6763e / ext4 errors=remoun
# /boot/efi was on /dev/sda2 during installation
UUID=6E9E-1A2A /boot/efi vfat umask=0077 0 1
/swapfile none swap sw
/dev/vg1/lv1 /mnt/ext4 ext4 defaults 0 2
/dev/vg1/lv2 /mnt/ext3 ext3 defaults 0 2

```

Figura 9 – Edição de ficheiro

### Solução:

Os sistemas de ficheiros foram montados nos diretórios /mnt/ext4 e /mnt/ext3. A persistência da montagem pode ser configurada adicionando as entradas adequadas no arquivo /etc/fstab, embora este passo não tenha sido abordado neste relatório.

### e) Criar um arquivo com permissões específicas:

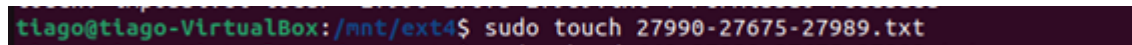
Objetivo: Criar um arquivo com permissões específicas dentro do sistema de arquivos montado.

Passo: No terminal, criamos o ficheiro dentro da diretoria /mnt/ext4 com o nome composto pelo grupo dos números dos alunos que realizaram o devido trabalho (22990-27675-27999.txt). De seguida, dar as devidas e respetivas permissões ao root, grupo e outros, sendo que root (leitura e escrita/ 6), grupo (nenhuma permissão/0) e outros (somente leitura/4).

Comandos:

1 - sudo touch /mnt/ext4/27990-27675-27989.txt

2 - sudo chmod 640 /mnt/ext4/27990-27675-27989.txt



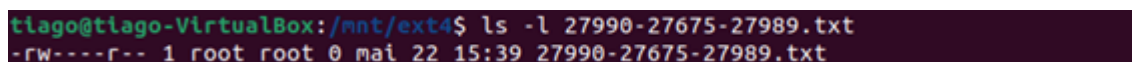
```
tiago@tiago-VirtualBox:/mnt/ext4$ sudo touch 27990-27675-27989.txt
```

Figura 10 – Criar ficheiro com nº de alunos



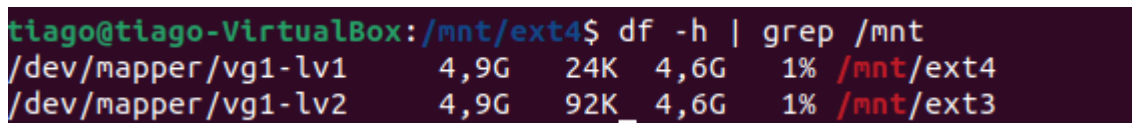
```
tiago@tiago-VirtualBox:/mnt/ext4$ sudo chmod 604 27990-27675-27989.txt
```

Figura 11 – Permissão



```
tiago@tiago-VirtualBox:/mnt/ext4$ ls -l 27990-27675-27989.txt
-rw----r-- 1 root root 0 mai 22 15:39 27990-27675-27989.txt
```

Figura 12 – Listagem de permissão



```
tiago@tiago-VirtualBox:/mnt/ext4$ df -h | grep /mnt
/dev/mapper/vg1-lv1      4,9G   24K   4,6G    1% /mnt/ext4
/dev/mapper/vg1-lv2      4,9G   92K   4,6G    1% /mnt/ext3
```

Figura 13 – Mostra o uso de espaço em disco dos sistemas de ficheiros montados em diretórios

Solução:

Foi criado um ficheiro chamado 27990-27675-27989.txt dentro do diretório /mnt/ext4, e suas permissões foram ajustadas para que o proprietário tenha permissões de leitura e escrita, o grupo não tenha permissões e outros usuários tenham apenas permissão de leitura.

## Parte B)

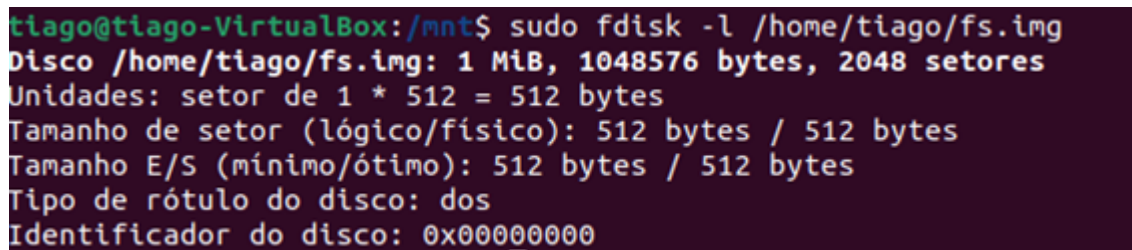
### a) Identificar o bloco de início dos ficheiros regulares presentes do sistema de ficheiros

Objetivo: Identificar o bloco de início dos ficheiros.

Passos: No terminal, executamos os comandos, “sudo fdisk -l /home/tiago/fs.img” que é usado para listar as partições e detalhes do ficheiro de imagem do sistema de arquivos especificado, comando “sudo fsstat /home/tiago/fs.img” que é usado para exibir estatísticas detalhadas sobre um sistema de ficheiros presente no arquivo de imagem “fs.img” e por fim o comando “sudo fls /home/tiago/fs.img” que é utilizado para listar ficheiros e diretórios presentes em uma imagem de sistema de arquivos, como o ficheiro “fs.img”.

Comandos:

- 1 - sudo fdisk -l /home/tiago/fs.img
- 2 - sudo fsstat /home/tiago/fs.img
- 3 - sudo fls /home/tiago/fs.img



```
tiago@tiago-VirtualBox:/mnt$ sudo fdisk -l /home/tiago/fs.img
Disco /home/tiago/fs.img: 1 MiB, 1048576 bytes, 2048 setores
Unidades: setor de 1 * 512 = 512 bytes
Tamanho de setor (lógico/físico): 512 bytes / 512 bytes
Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes
Tipo de rótulo do disco: dos
Identificador do disco: 0x00000000
```

Figura 14 – Info do fs.img



```

tiago@tiago-VirtualBox:/mnt$ sudo fsstat /home/tiago/fs.img
FILE SYSTEM INFORMATION
-----
File System Type: FAT12

DEM Name: mkfs.fat
Volume ID: 0xaa724ea
Volume Label (Boot Sector): NO NAME
Volume Label (Root Directory):
File System Type Label: FAT12

Sectors before file system: 0

File System Layout (in sectors)
Total Range: 0 - 2047
* Reserved: 0 - 0
** Boot Sector: 0
* FAT 0: 1 - 2
* FAT 1: 3 - 4
* Data Area: 5 - 2047
** Root Directory: 5 - 36

```

Figura 15 – Comando fsstat para info do sistema

```

** Cluster Area: 37 - 2044
** Non-clustered: 2045 - 2047

METADATA INFORMATION
-----
Range: 2 - 32694
Root Directory: 2

CONTENT INFORMATION
-----
Sector Size: 512
Cluster Size: 2048
Total Cluster Range: 2 - 503

FAT CONTENTS (in sectors)
-----
65-68 (4) -> EOF

```



Figura 16 – Fim da informação do sistema (restante continuação de figura anterior)

```
tiago@tiago-VirtualBox:/mnt$ sudo fls /home/tiago/fs.img
r/r * 4:      ipca.txt
r/r * 5:      _ICHEI~1.SWP
r/r 7:  ficheiro1
r/r * 9:      ficheiro2
r/r * 10:     _ICHEI~2.SWP
v/v 32691:    $MBR
v/v 32692:    $FAT1
v/v 32693:    $FAT2
v/v 32694:    $OrphanFiles
```

Figura 17 – Listar inodes dos ficheiros

**b) Indique o conteúdo do ficheiro com o nome ipca.txt.**

Objetivo: Indicar conteúdo do ficheiro.

Passos: No terminal, executar o comando “sudo icat /home/tiago/fs.img 4”. Deste modo irá emitir o composto conteúdo do ficheiro.

Comandos:

1 - sudo icat /home/tiago/fs.img

```
tiago@tiago-VirtualBox:/mnt$ sudo icat /home/tiago/fs.img 4
Este é o conteúdo do ficheiro ipca.txt
```

Figura 17 – Apresentação do conteúdo de “ipca.txt”

## Conclusão

Concluindo, este trabalho prático permitiu-nos aplicar os conceitos teóricos de gestão de processos e ficheiros em um contexto prático. Através da implementação de ferramentas e da análise de sistemas de ficheiros, consolidamos o nosso conhecimento e desenvolvemos habilidades essenciais para a nossa futura carreira em Engenharia Informática.

# Bibliografia

[31 comandos básicos, mas essenciais do Ubuntu \(linux-console.net\)](http://linux-console.net)

moodle - disciplina sistemas operativos (PowerPoints)

[Manual do Ubuntu - Página Inicial \(ubuntu-manual.org\)](http://ubuntu-manual.org)

[Geeks for Geeks - Introdução às chamadas ao sistema](#)