

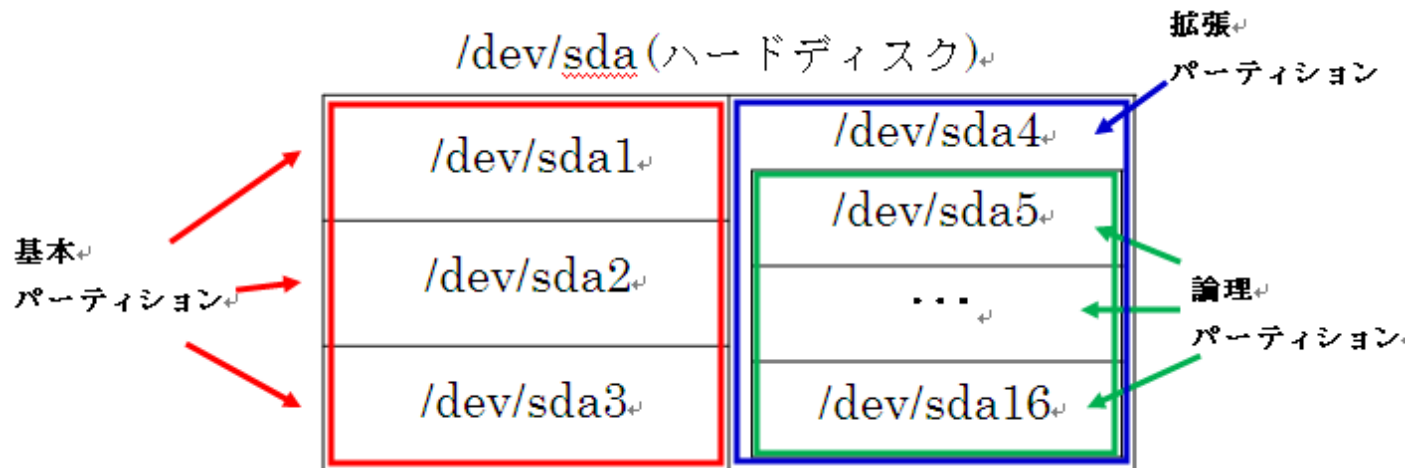
ファイルシステム

(1)パーティション

システム全体で5個以上のパーティションが必要になる場合は、まず拡張領域を作成し、その中に論理パーティションを作成していきます。これは、基本領域はシステム全体で4個までしか作成できない為だからです。

そこで/dev/sda4 という拡張領域を作成し、それに/dev/sda5 の論理パーティションを割り当てます。

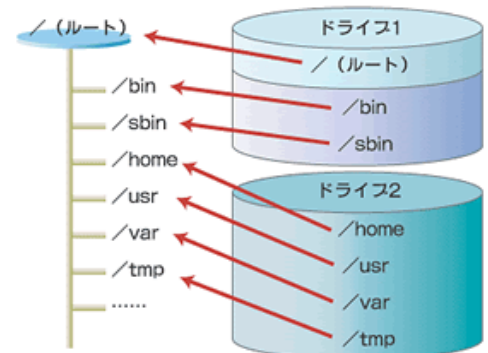
論理パーティションは/dev/sda16 まで作成可能なので、基本領域を含め最大15個のパーティションが作成できます。



※VMware の方は「/dev/nvme0n1p1～p16」に読み替えて下さい。

※UTM の方ではパーティションは128(!)まで作成できます

/dev/sda1 ～sda3 : 基本パーティション
 /dev/sda4 : 拡張パーティション
 /dev/sda5 ～sda16 : 論理パーティション



※今日の実習は先生の話と pdf 資料をしっかりと確認して下さい！

中途半端にしていると OS を破壊してしまいます！！ ←マジですよ (^_^;)

※誤って壊したら「NT43 Linux 資料 15 OS のバックアップとリカバリ (6月5日)」に従って Linux のリカバリして下さい！

※UTM を使用している方は P13 参照

※VirtualBox を使用している方は以下の作業では「/dev/nvme0n1」→「/dev/sda」に読み替え

(2)fdisk による新規パーティションの作成

```
[hal@motobe ~]$ df
```

ファイルシス	1K-ブロック	使用	使用可	使用%	マウント位置
devtmpfs	4096	0	4096	0%	/dev
tmpfs	896380	0	896380	0%	/dev/shm
tmpfs	358556	7344	351212	3%	/run
/dev/nvme0n1p3	12277760	5240460	7037300	43%	/
/dev/nvme0n1p1	1038336	393060	645276	38%	/boot
tmpfs	179276	96	179180	1%	/run/user/1000

1) 以下のコマンドを実行

```
[hal@motobe ~]$ sudo fdisk /dev/nvme0n1
```

fdisk (util-linux 2.37.4) へようこそ。

ここで設定した内容は、書き込みコマンドを実行するまでメモリのみに保持されます。

書き込みコマンドを使用する際は、注意して実行してください。

This disk is currently in use - repartitioning is probably a bad idea.

It's recommended to umount all file systems, and swapoff all swap partitions on this disk.

2) fdisk プログラムのプロンプトが表示される。パーティション情報を確認

コマンド (m でヘルプ): **p**

ディスク /dev/nvme0n1: 30 GiB, 32212254720 バイト, 62914560 セクタ

ディスク型式: VMware Virtual NVMe Disk

単位: セクタ (1 * 512 = 512 バイト)

セクタサイズ (論理 / 物理): 512 バイト / 512 バイト

I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト

ディスクラベルのタイプ: dos

ディスク識別子: 0x62c17c9f

デバイス	起動	開始位置	終了位置	セクタ	サイズ	Id	タイプ
/dev/nvme0n1p1 *		2048	2099199	2097152	1G	83	Linux
/dev/nvme0n1p2		2099200	6293503	4194304	2G	82	Linux スワップ / Solaris
/dev/nvme0n1p3		6293504	30869503	24576000	11.7G	83	Linux

3) fdisk の終了

コマンド (m でヘルプ): **q**

4) 新しいパーティションを作成

[hal@motobe ~]\$ **sudo fdisk /dev/nvme0n1**

コマンド (m でヘルプ): **p**

デバイス	起動	開始位置	終了位置	セクタ	サイズ	Id	タイプ
/dev/nvme0n1p1 *		2048	2099199	2097152	1G	83	Linux
/dev/nvme0n1p2		2099200	6293503	4194304	2G	82	Linux スワップ / Solaris
/dev/nvme0n1p3		6293504	30869503	24576000	11.7G	83	Linux

※現在のパーティション構成確認

①拡張パーティション作成

コマンド (m でヘルプ): **n**

パーティションタイプ

- p 基本パーティション (3 プライマリ, 0 拡張, 1 空き)
- e 拡張領域 (論理パーティションが入ります)

選択 (既定値 e): **e** ←パーティションタイプ (e : 拡張 (extended)) を入力

パーティション 4 を選択

最初のセクタ (30869504-62914559, 既定値 30869504): ←そのままエンター

最終セクタ, +/-セクタ番号 または +/-サイズ{K,M,G,T,P} (30869504-62914559, 既定値 62914559):
←そのままエンター

新しいパーティション 4 をタイプ Extended、サイズ 15.3 GiB で作成しました。

コマンド (m でヘルプ): **p**

デバイス	起動	開始位置	終了位置	セクタ	サイズ	Id	タイプ
/dev/nvme0n1p1 *		2048	2099199	2097152	1G	83	Linux
/dev/nvme0n1p2		2099200	6293503	4194304	2G	82	Linux スワップ / Solaris
/dev/nvme0n1p3		6293504	30869503	24576000	11.7G	83	Linux
/dev/nvme0n1p4		30869504	62914559	32045056	15.3G	5	拡張領域

※空き領域全部を拡張領域に設定。また、表示が拡張であることを確認。

②論理パーティション作成

コマンド (m でヘルプ): **n**

全ての基本パーティションが使用済みです。

論理パーティション 5 を追加します

最初のセクタ (30871552-62914559, 既定値 30871552):

←そのままエンター

最終セクタ, +/-セクタ番号 または +/-サイズ{K,M,G,T,P} (30871552-62914559, 既定値 62914559): **+2G**

※今回は新しいパーティションとして「2GB」作成

新しいパーティション 5 をタイプ Linux、サイズ 2 GiB で作成しました。

コマンド (m でヘルプ): **p**

デバイス	起動	開始位置	終了位置	セクタ	サイズ	Id	タイプ
/dev/nvme0n1p1 *		2048	2099199	2097152	1G	83	Linux
/dev/nvme0n1p2		2099200	6293503	4194304	2G	82	Linux スワップ / Solaris
/dev/nvme0n1p3		6293504	30869503	24576000	11.7G	83	Linux
/dev/nvme0n1p4		30869504	62914559	32045056	15.3G	5	拡張領域
/dev/nvme0n1p5		30871552	35065855	4194304	2G	83	Linux

※論理パーティション 2GB を作成

※さらに追加する場合は、上記の ②を繰り返し実行する。

③空き容量の確認

コマンド (m でヘルプ): **F**

開始位置 終了位置 セクタ サイズ

35067904 62914559 27846656 13.3G

④設定の保存

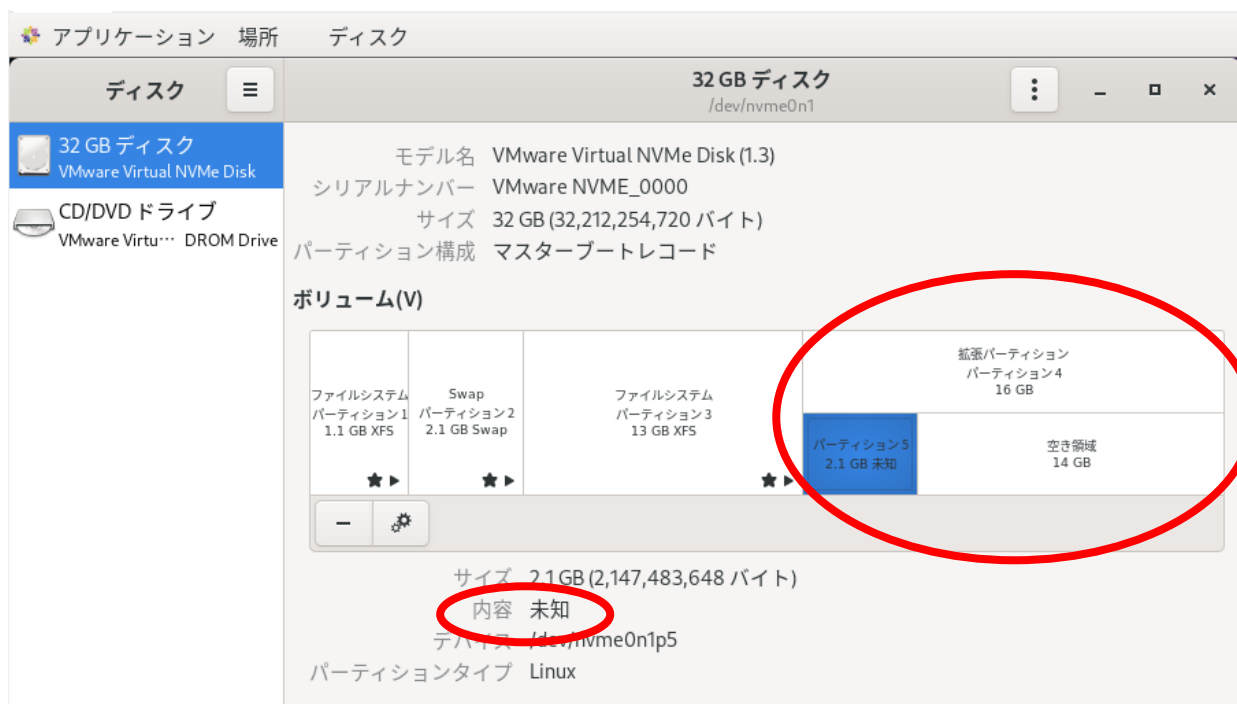
コマンド (m でヘルプ): **w**

パーティション情報が変更されました。

ディスクを同期しています。

※失敗したり、やり直しは **q** でキャンセル

※書き込みが終了すると、端末: [hal@motobe ~]\$ に戻る。



※Linux のメニュー → ディスク で確認

※パーティション 5 を選択 → 「内容 未知」 に注目！

```
[hal@motobe ~]$ ls /dev/nvme0n1
```

```
nvme0n1 nvme0n1p1 nvme0n1p2 nvme0n1p3 nvme0n1p4 nvme0n1p5
```

※一覧の中に nvme0n1p4 (拡張パーティション)、nvme0n1p5(論理パーティション)が有ることの確認

5)フォーマット

作成したパーティションを Linux で使用出来るファイル形式(ext4)に変換(フォーマット)

```
[hal@motobe ~]$ sudo mkfs -t ext4 /dev/nvme0n1p5
```

```
mke2fs 1.46.5 (30-Dec-2021)
```

```
Creating filesystem with 524288 4k blocks and 131072 inodes
```

```
Filesystem UUID: 4a6e6ea0-101f-4c1e-ada2-25387eb84b19
```

```
Superblock backups stored on blocks:
```

```
32768, 98304, 163840, 229376, 294912
```

```
Allocating group tables: done
```

```
Writing inode tables: done
```

```
Creating journal (16384 blocks): done
```

```
Writing superblocks and filesystem accounting information: done
```

※自動で進みます

6)マウント

①マウントポイントを/test とする ←あらかじめ、/test を作成しておくこと

```
[hal@motobe ~]$ sudo mkdir /test
```

```
[hal@motobe ~]$ df
```

ファイルシス	1K-ブロック	使用	使用可	使用%	マウント位置
devtmpfs	4096	0	4096	0%	/dev
tmpfs	896380	0	896380	0%	/dev/shm
tmpfs	358556	7352	351204	3%	/run
/dev/nvme0n1p3	12277760	5238848	7038912	43%	/
/dev/nvme0n1p1	1038336	393060	645276	38%	/boot
tmpfs	179276	100	179176	1%	/run/user/1000

※まだマウントしていないので表示されません

```
[hal@motobe ~]$ ls /test/
```

※中身はまだ空

②マウント設定

```
[hal@motobe ~]$ sudo mount /dev/nvme0n1p5 /test/
```

```
[hal@motobe ~]$ df
```

ファイルシス	1K-ブロック	使用	使用可	使用%	マウント位置
devtmpfs	4096	0	4096	0%	/dev
tmpfs	896380	0	896380	0%	/dev/shm
tmpfs	358556	7364	351192	3%	/run
/dev/nvme0n1p3	12277760	5238900	7038860	43%	/
/dev/nvme0n1p1	1038336	393060	645276	38%	/boot
tmpfs	179276	100	179176	1%	/run/user/1000
/dev/nvme0n1p5	1992552	24	1871288	1%	/test

※マウント出来ました！

```
[hal@motobe ~]$ mount
```

(省略)

```
/dev/nvme0n1p5 on /test type ext4 (rw,relatime)
```

```
[hal@motobe ~]$ ls /test/
```

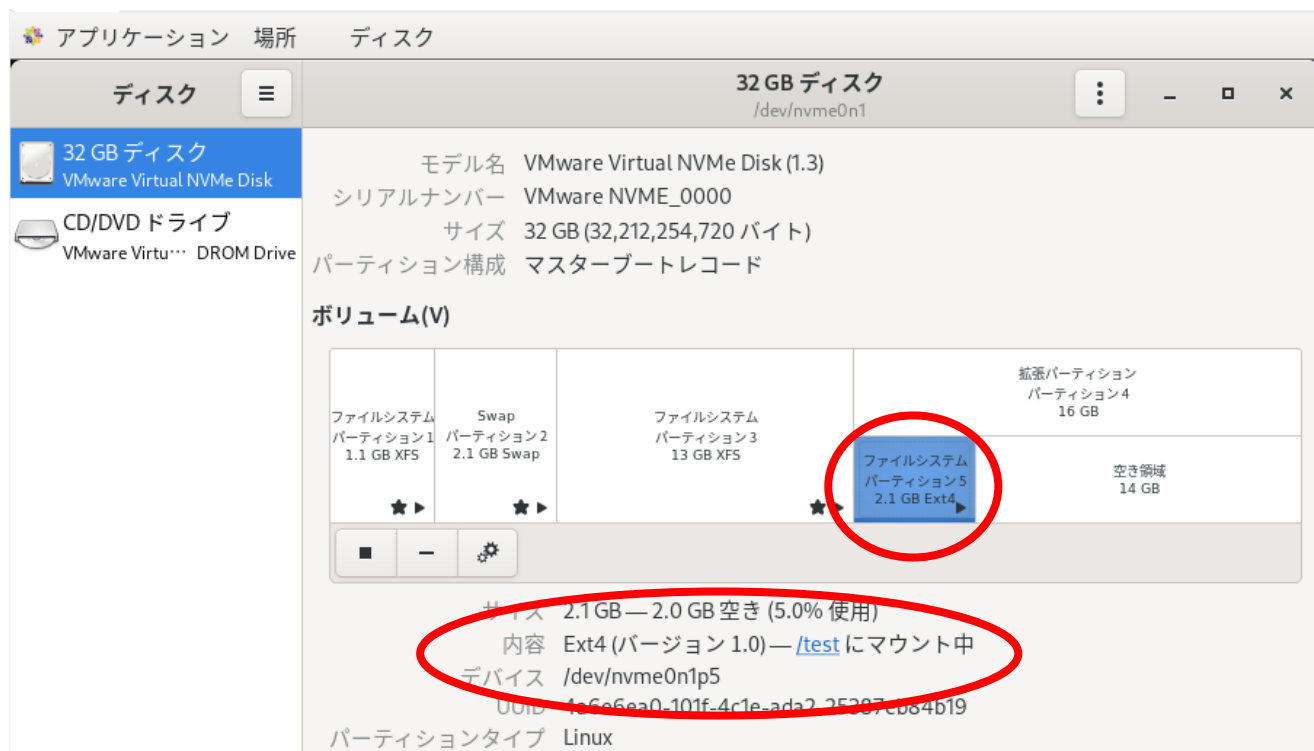
lost+found

※マウント以前は無かった「lost+found」(これは別パーティションの時に出来る)が存在

```
[hal@motobe ~]$ sudo touch /test/abc.txt ←ファイルを作成(使用)できるか
```

```
[hal@motobe ~]$ ls /test/
```

abc.txt lost+found



③マウントを外す

- ・アンマウントする時には必ずマウンポイントから出ること！

```
[hal@motobe ~]$ cd /test/ ←今回だけわざとマウントポイントに移動
```

```
[hal@motobe test]$ pwd
```

```
/test
```

```
[hal@motobe test]$ sudo umount /test ←アンマウント
```

umount: /test: 対象は使用中です. ←windows や MAC でも良く見るエラーですね

```
[hal@motobe test]$ cd ←マウントポイントから出る
```

```
[hal@motobe ~]$ sudo umount /test
```

```
[hal@motobe ~]$ df
```

ファイルシス	1K-ブロック	使用	使用可	使用%	マウント位置
devtmpfs	4096	0	4096	0%	/dev
tmpfs	896380	0	896380	0%	/dev/shm
tmpfs	358556	7356	351200	3%	/run
/dev/nvme0n1p3	12277760	5238904	7038856	43%	/
/dev/nvme0n1p1	1038336	393060	645276	38%	/boot
tmpfs	179276	100	179176	1%	/run/user/1000

```
[hal@motobe ~]$ mount
```

※勿論、/test も見えない

```
[hal@motobe ~]$ ls /test/
```

※アンマウントした為、別パーティションに有る abc.txt は見えなくなり、空のマウントポイントのみ存在

※lost+found も消える

7)ファイルシステムのチェック

長く Linux を使用するとファイルシステムのエラーが溜まってきますのでチェック

```
[hal@motobe ~]$ sudo fsck /dev/nvme0n1p5
```

```
fsck from util-linux 2.37.4
```

```
e2fsck 1.46.5 (30-Dec-2021)
```

```
/dev/nvme0n1p5: clean, 12/131072 files, 26156/524288 blocks
```

(3)パーティションの削除

```
[hal@motobe ~]$ sudo fdisk /dev/nvme0n1
```

コマンド (m でヘルプ): **p** ←現状確認(削除するパーティションの確認)

デバイス	起動	開始位置	終了位置	セクタ	サイズ	Id	タイプ
/dev/nvme0n1p1 *		2048	2099199	2097152	1G	83	Linux
/dev/nvme0n1p2		2099200	6293503	4194304	2G	82	Linux


```

/dev/nvme0n1p3      6293504 30869503 24576000  11.7G 83 Linux
/dev/nvme0n1p4      30869504 62914559 32045056  15.3G  5 拡張領域
/dev/nvme0n1p5      30871552 35065855  4194304    2G 83 Linux

```

コマンド (m でヘルプ): **d** ←削除

パーティション番号 (1-5, 既定値 5): **5** ←削除したいパーティション番号

パーティション 5 を削除しました。

コマンド (m でヘルプ): **p**

```

デバイス      起動 開始位置 終了位置   セクタ サイズ Id タイプ
/dev/nvme0n1p1 *      2048  2099199  2097152    1G 83 Linux
/dev/nvme0n1p2      2099200  6293503  4194304    2G 82 Linux スワップ
/dev/nvme0n1p3      6293504 30869503 24576000  11.7G 83 Linux
/dev/nvme0n1p4      30869504 62914559 32045056  15.3G  5 拡張領域

```

※簡単に論理パーティションが消えましたね！

コマンド (m でヘルプ): **d**

パーティション番号 (1-4, 既定値 4): **4**

パーティション 4 を削除しました。

コマンド (m でヘルプ): **p**

```

デバイス      起動 開始位置 終了位置   セクタ サイズ Id タイプ
/dev/nvme0n1p1 *      2048  2099199  2097152    1G 83 Linux
/dev/nvme0n1p2      2099200  6293503  4194304    2G 82 Linux スワップ
/dev/nvme0n1p3      6293504 30869503 24576000  11.7G 83 Linux

```

※簡単に拡張パーティションも消えましたね！

※大丈夫と思いますが、絶対にパーティション 1、2、3 は削除しないで下さい！Linux が動かなくなります！

コマンド (m でヘルプ): **w**

パーティション情報が変更されました。

パーティション 5 の削除に失敗しました: デバイスもしくはリソースがビジー状態です

カーネルは古い情報を使用しています。新しい情報はシステムを再起動した後に利用されます。

ディスクを同期しています。

[hal@motobe ~]\$ **sudo reboot**

●再起動後

```
[hal@motobe ~]$ sudo fdisk /dev/nvme0n1
```

コマンド (m でヘルプ): **p**

```

デバイス      起動 開始位置 終了位置   セクタ サイズ Id タイプ
/dev/nvme0n1p1 *          2048  2099199  2097152    1G 83 Linux
/dev/nvme0n1p2          2099200  6293503  4194304    2G 82 Linux スワップ / Solari
/dev/nvme0n1p3          6293504 30869503 24576000  11.7G 83 Linux
※パーティションが削除されています

```

コマンド (m でヘルプ): **q**

【実習】新しいパーティションの作成

(変更前)

```

/dev/nvme0n1p1 *      /boot
/dev/nvme0n1p2        swap
/dev/nvme0n1p3        /

```

(変更後)

```

/dev/nvme0n1p1 *      /boot
/dev/nvme0n1p2        swap
/dev/nvme0n1p3        /
/dev/nvme0n1p4        (拡張領域)
/dev/nvme0n1p5        /work

```

1) 上記(変更後)のように nvme0n1p5 を **3GB** で作成

●もし以下のメッセージが出たら

新しいパーティション 5 をタイプ Linux、サイズ 3 GiB で作成しました。
パーティション #5 には ext4 署名が書き込まれています。

署名を削除しますか? [Y]es/[N]o: **Y** ←署名を削除

署名は write (書き込み)コマンドを実行すると消えてしまいます

- 2) 新規に追加したパーティションに対して、**ext4** ファイルシステムのフォーマットをする。
- 3) **マウントポイント(/work)**を作成する。
- 4) 追加したパーティションのファイルシステムのチェックを行う。
- 5) **/dev/nvme0n1p5 に/work をタイプ ext4 でマウント**
- 6) 各種、確認

★★以下の/etc/fstab の編集を間違えると**最悪、Linux が壊れます！！**←マジです！

7)再起動

再起動すると、5)でマウントした**/work** のマウントが外れている。

→ 再起動の度に毎回 **mount** コマンドを使用しなくてはならない！ →面倒！

[hal@motobe ~]\$ **df**

ファイルシス	1K-ブロック	使用	使用可	使用%	マウント位置
devtmpfs	4096	0	4096	0%	/dev
tmpfs	896380	0	896380	0%	/dev/shm
tmpfs	358556	7348	351208	3%	/run
/dev/nvme0n1p3	12277760	5239240	7038520	43%	/
/dev/nvme0n1p1	1038336	393060	645276	38%	/boot
tmpfs	179276	96	179180	1%	/run/user/1000

8) /etc/fstab を編集する (/dev/nvme0n1p5 に/work をマウントする場合)

[hal@motobe ~]\$ **sudo cp /etc/fstab /etc/fstab.old**

※あらかじめ設定ファイルをバックアップ

[hal@motobe ~]\$ **sudo vi /etc/fstab**

以下の行を最終行に追記し、セーブする。(vi エディタの使い方は覚えていますよね?)

/dev/sda5 /work ext4 defaults 1 2

```
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=ed4ac125-292b-452e-935c-c4a83b27b98c / xfs defaults 0 0
UUID=121db107-70f5-4c2a-8bae-b24130ff1508 /boot xfs defaults 0 0
UUID=1878e553-dc1c-4419-8fc8-706e67af7a10 none swap defaults 0 0
/dev/nvme0n1p5 /work ext4 defaults 1 2
```

9) /etc/fstab の記述を利用してマウントできることを確認する。(p312)

```
[hal@motobe ~]$ sudo mount -a
```

※fstab に載っている設定を反映

```
[hal@motobe ~]$ df
```

ファイルシス	1K-ブロック	使用	使用可	使用%	マウント位置
devtmpfs	4096	0	4096	0%	/dev
tmpfs	896380	0	896380	0%	/dev/shm
tmpfs	358556	7356	351200	3%	/run
/dev/nvme0n1p3	12277760	5239356	7038404	43%	/
/dev/nvme0n1p1	1038336	393060	645276	38%	/boot
tmpfs	179276	96	179180	1%	/run/user/1000
/dev/nvme0n1p5	3021608	24	2847916	1%	/work

10) 再起動後、新規パーティション(/dev/nvme0n1p5)が自動でマウントされていることを確認する。

```
[hal@motobe ~]$ sudo reboot
```

※再起動後

```
[hal@motobe ~]$ df
```

ファイルシス	1K-ブロック	使用	使用可	使用%	マウント位置
devtmpfs	4096	0	4096	0%	/dev
tmpfs	896380	0	896380	0%	/dev/shm
tmpfs	358556	7344	351212	3%	/run
/dev/nvme0n1p3	12277760	5240404	7037356	43%	/
/dev/nvme0n1p1	1038336	393060	645276	38%	/boot
/dev/nvme0n1p5	3021608	24	2847916	1%	/work
tmpfs	179276	96	179180	1%	/run/user/1000

```
[hal@motobe ~]$ mount
```

```
/dev/nvme0n1p5 on /work type ext4 (rw,relatime)
```

```
[hal@motobe ~]$ ls /work/
```

```
lost+found
```

※自動でマウントされています

以上！

◎UTM を使用している方へ

最初のパーティション分割の方法が違いますのでスクリーンショットを張ります。



※デバイスが vda になっているのが大きな違いです ←重要！！

※以下のコマンドは「sudo ~」で実行して下さい

```

root@motobe:~
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
[hal@motobe ~]$ su -
パスワード:
最終ログイン: 2022/10/15 (土) 17:11:02 JST日時 pts/0
[root@motobe ~]# df
ファイル システム      1K-ブロック   使用   使用可   使用%   マウント位置
devtmpfs                773760      0   773760     0%   /dev
tmpfs                   792064      0   792064     0%   /dev/shm
tmpfs                   792064   36032   756032     5%   /run
tmpfs                   792064      0   792064     0%   /sys/fs/cgroup
/dev/mapper/centos-root 15718400 4733648 10984752    31%   /
/dev/vda2               1038336  221416   816920    22%   /boot
/dev/vda1               2045808    9264   195316     5%   /boot/efi
tmpfs                   158464      64   158400     1%   /run/user/42
tmpfs                   158464     384   158080     1%   /run/user/1000
[root@motobe ~]# ls /dev/vda
vda vda1 vda2 vda3
[root@motobe ~]# ls /dev/vda

```

※現在のパーティションは3つ(vda1、vda2、vda3)あります。

```

root@motobe:~
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
[root@motobe ~]# fdisk /dev/vda
WARNING: fdisk GPT support is currently new, and therefore in an experimental phase. Use at your own discretion.
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

コマンド (m でヘルプ): p

Disk /dev/vda: 32.2 GB, 32212254720 bytes, 62914560 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト
Disk label type: gpt
Disk identifier: A8D3A802-E976-444F-9778-551F4A3D0240

#       Start       End     Size Type System      Name
#  -----
1         2048       411647   200M  EFI System      EFI System Partition
2      411648     2508799    1G  Microsoft basic
3     2508800     38172671   17G  Linux LVM

```

```

コマンド (m でヘルプ): n
パーティション番号 (4-128, default 4):
First sector (34-62914526, default 38172672):
Last sector, +sectors or +size(K,M,G,T,P) (38172672-62914526, default 62914526): +3G
Created partition 4

```

```

コマンド (m でヘルプ): p

Disk /dev/vda: 32.2 GB, 34359738368 bytes, 62914560 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト
Disk label type: gpt
Disk identifier: A8D3A802-E976-444F-9778-551F4A3D0240

#      Start      End      Size Type      Name
#  -----
1      2048      411647    200M EFI System EFI System Partition
2      411648    2508800    16G Microsoft basic
3      2508800    38172671   17G Linux LVM
4      38172672    44464127   3G Linux filesystem

```

※4 つ目のパーティション(vda4)が 3GB で出来ました！

```

コマンド (m でヘルプ): w
パーティションテーブルは変更されました！

ioctl() を呼び出してパーティションテーブルを再読み込みします。

WARNING: Re-reading the partition table failed with error 16: デバイスもしくはリソースがビジー状態です。
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
ディスクを同期しています。
[root@motobe ~]# reboot

```

[hal@motobe ~]\$ **sudo mkfs -t ext4 /dev/vda4**

```

root@motobe:~
mkfs 1.42.9 (28-Dec-2013)
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
196608 inodes, 786432 blocks
39321 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=805306368
24 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

[root@motobe ~]#

```

後の流れは同じです

```

[root@motobe ~]# mkdir /work
[root@motobe ~]# ls /work/
[root@motobe ~]# mount /dev/vda4 /work/
[root@motobe ~]# ls /work/
lost+found
[root@motobe ~]# df
ファイルシステム      1K-ブロック   使用   使用可   使用%   マウント位置
devtmpfs                773696         0    773696     0%   /dev
tmpfs                   792000         0    792000     0%   /dev/shm
tmpfs                   792000    35968    756032     5%   /run
tmpfs                   792000         0    792000     0%   /sys/fs/cgroup
/dev/mapper/centos-root 15718400 4732736 10985664    31%   /
/dev/vda2              1038336    221416    816920    22%   /boot
/dev/vda1               204580      9264    195316     5%   /boot/efi
tmpfs                  158464         0    158464     1%   /run/user/42
tmpfs                  158464    320     158144     1%   /run/user/1000
/dev/vda4              3030800     9216    2847916     1%   /work
[root@motobe ~]# touch /work/abc.txt
[root@motobe ~]# ls /work/
abc.txt  lost+found
[root@motobe ~]#

```

※新しいパーティション(vda4)を/work でマウントしました。



以上！