

学 生 工 学 実 験

2-1 並列分散処理

1週目 Mpich (pp. 4-37)

2週目 Hadoop (pp. 38-55)

2 – 1 分散ネットワーク

目的

- ・並列分散環境の構築に必要なネットワーク構成を理解し、自分で構築する。
- ・並列分間環境におけるデータ処理を通して、ビッグデータ分析技術を体験し、理解する。

概要

IoT の普及並びに計測技術の発展に伴い、周囲環境のセンシングデータが活用される事例が増加している。また、インターネットを介して我々は非常に多くの情報を容易に取得することが可能になってきた。多様なデバイスから得た大規模なデータは、即時的に解析され、実社会における課題解決のために活用されることが望ましい。このため、大規模データを素早く解析するための高速化技術は、近年特に重要視されている。本実験の第 1 回は、まず仮想環境を用いるため VirtualBox をインストールし、その中で CentOS のインストール・セットアップを実施する。そして、本実験で用いる分散処理技術「Hadoop」のインストールを実施し、「Hadoop」の環境の構築を行う。また、第 2 回では、実際にインストールした「Hadoop」の環境を用いて、分散処理の実行に取り組む。以上の作業を通して、計算資源を有効に利用する分散処理について理解を進める。

実験構成

本実験でのネットワーク構成図を図 1 に示す。図 1 に示すように、192.168.11.1/24 のネットワークに学生 PC 内に設置する仮想マシン(slaveXXX)を全員接続し、全体を統括する master の下で全ての分散ネットワーク処理を実行する。

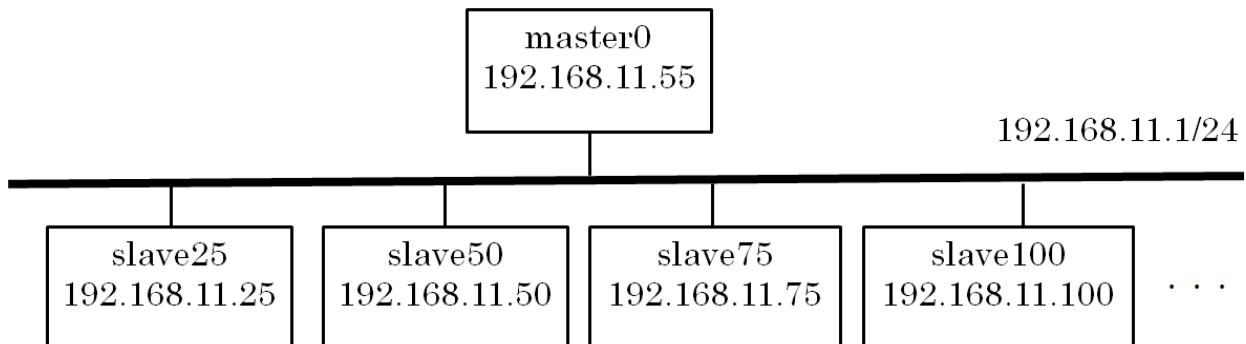


図 1 : 実験ネットワーク構成図

実験環境の構成は図 2 のように行う。

図 2 : 実験環境

実験での名称

- ・チーム：実験用 PC 1 台(master)を共有している 2 名ないし 3 名の学生
- ・グループ：同じ机（有線机か無線机）で受講する 4 名ないし 5 名の学生
- ・班：この実験を一緒に受講する 9 名ないし 10 名の学生

実験での割当て

- ・実験用 PC の M1 には S1,S2 および S9

T-1.(TA 向けの準備) 授業前にすること

実験環境の構築

ローカルネットワークへの接続

/home/tpu/share の中のファイルの削除

master 同士で ping の確認

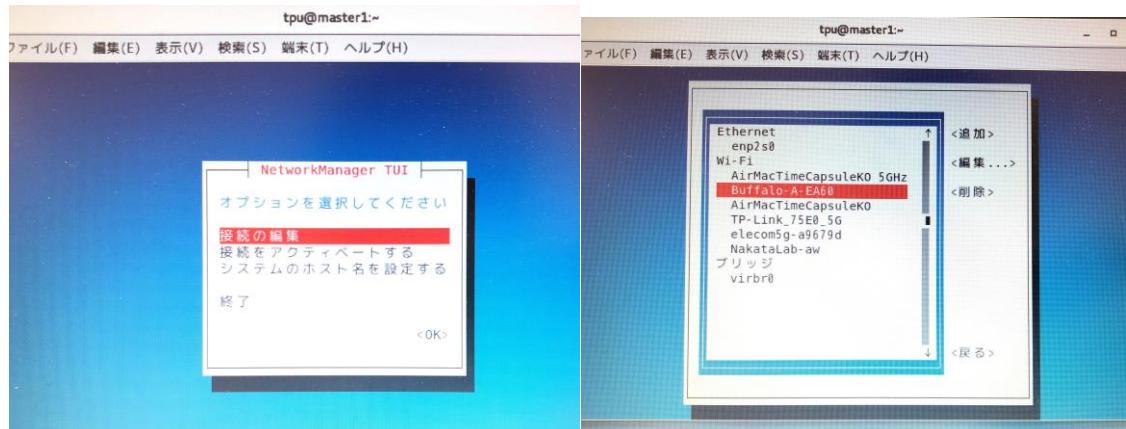
○master の無線 IP アドレス変更(確認)手順(→使用するルータが変わったときなど)

0)master を新しい Wi-Fi に接続する

1) 「端末」で「ifconfig」を入力し、「inet」の部分が 1~3 行目までルータと一致する(外のコマンドプロンプト「ipconfig」の Wireless LAN adapter Wi-Fi と 3 行目まで同じ)アドレスを控えておく。

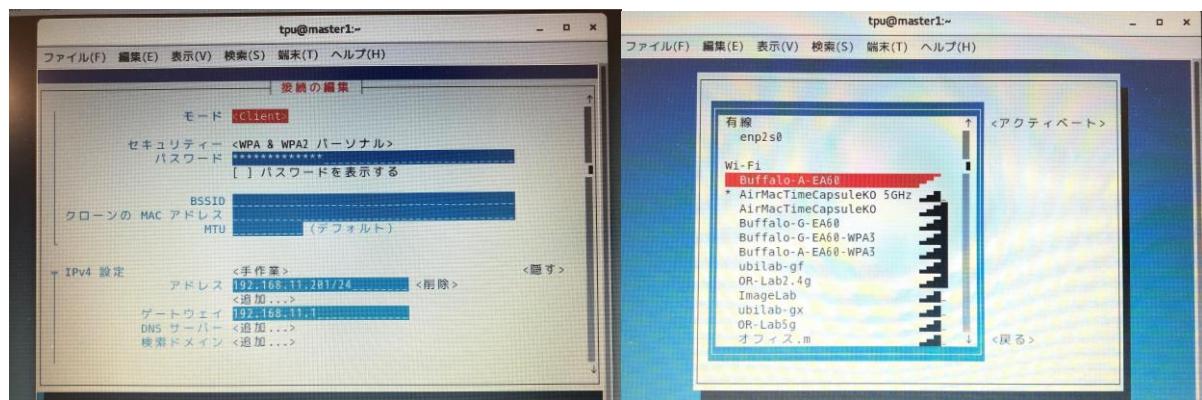
2) 同様に「nmtui」を入力

3) 接続の編集を Enter で選択



4) Wi-Fi で使用するルータ名を選択(↑ここでは Buffalo-A-EA60 の例)

5) IPv4 の項目を表示し、一番上は<手作業>、「アドレス」は 1)と同じもの、「ゲートウェイ」は 1~3 行目まではアドレスと同じで、最後だけ「X.X.X.1」となるように設定し、1 番下の OK を選択。



6) 1 番最初の画面に行き、「接続をアクティベートする」に行く。

7) 使用するルータ名を選択して<アクティベート>を Enter する。

(元々その状態なら<解除>(ルータ再表示までに時間がかかる)からアクティベート)

8) 最初の画面に戻って終了を選択する。

T-1.(TA 向けの準備)授業前にすること (終わり)

1週目

1週目では、Mpichにより、モンテカルロ法を用いて円周率を求める。

実験手順

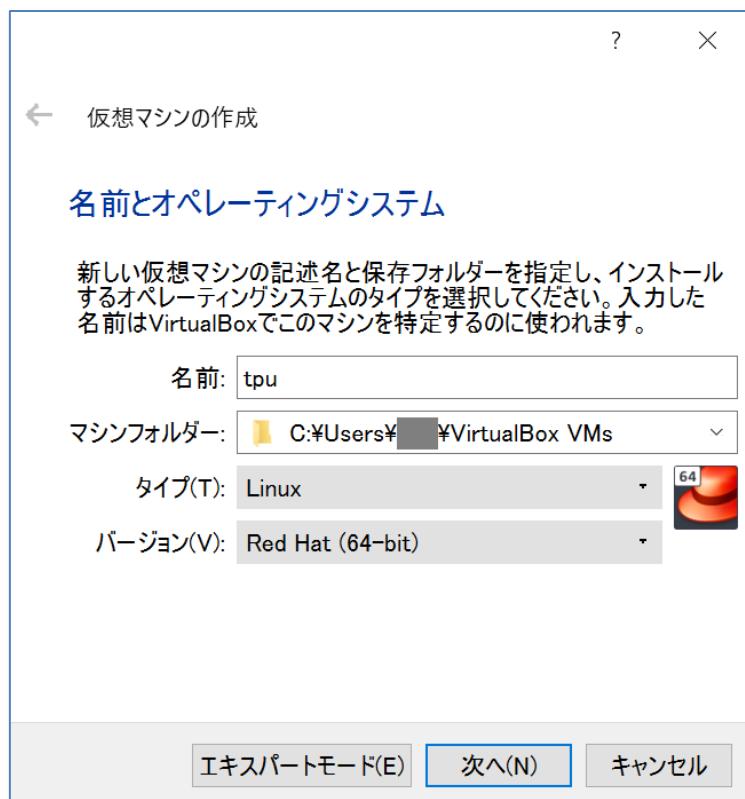
分散ネットワーク環境の構築(1週目)

1. まず2グループ(2つの机)に分かれ、着席する。(グループ分けは出欠状況によるため、当日調整する)
2. 2つのUSBメモリを配布する。それぞれの中から、「CentOS-7~」と「VirtualBox~」のファイルを、自分のPCにコピーする。
3. 「VirtualBox-6.1.16-140961-Win.exe」を実行し、VirtualBoxをインストールする。
※数字はバージョンが変わっている可能性がある(6.1.16は2020/10/16時点で最新)
4. VirtualBoxを起動する。
5. 左上の「新規」から仮想マシンを作成する。



6. 仮想マシンの名前は何でも良いがとりあえず「tpu」とする。タイプは「Linux」、バージョンは「Red Hat(64-bit)」を選択して次に進む。(CentOS は RedHat 系の OS である)

32bit しか表示されない場合、BIOS を開いて virtualization technology が Disable になっている場合に Enable に変更する。



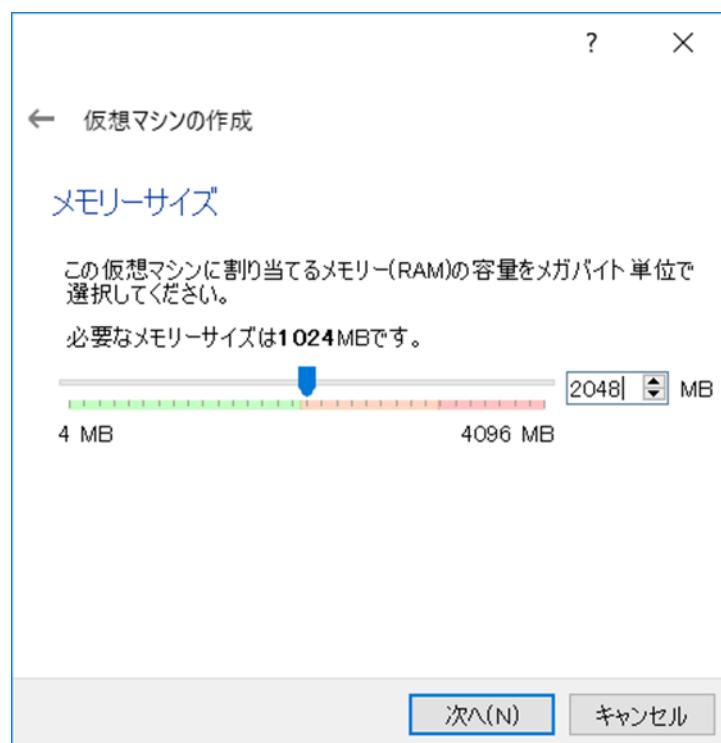
E-1. Red Hat(64-bit)が選択できない場合

PC の BIOS の画面に行き Intel Virtualization Technology を enable にする。

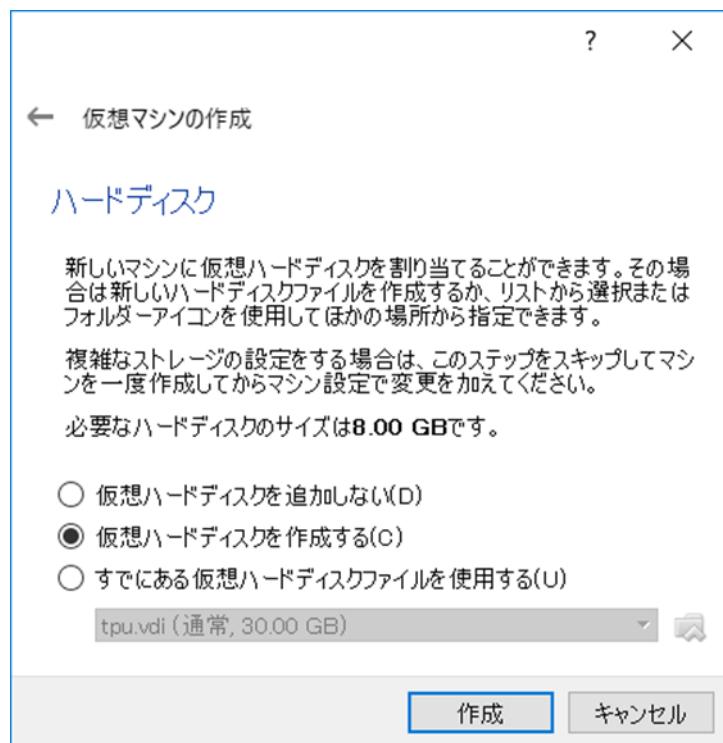
その後、もう一度起動すると選択できるようになる。

E-1. Red Hat(64-bit)が選択できない場合(終わり)

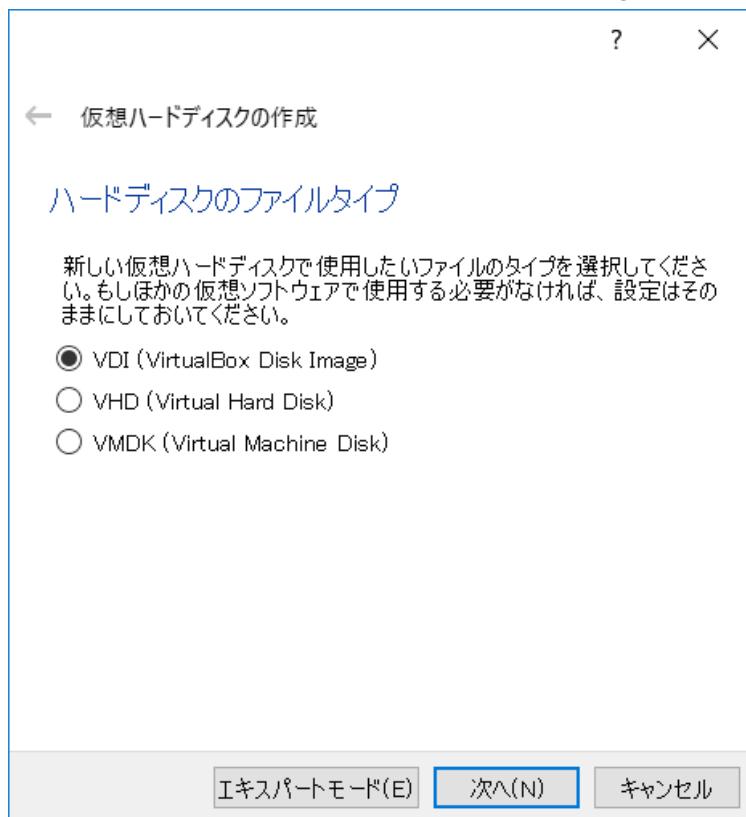
7. メモリーサイズは「2048MB」に設定して次に進む。



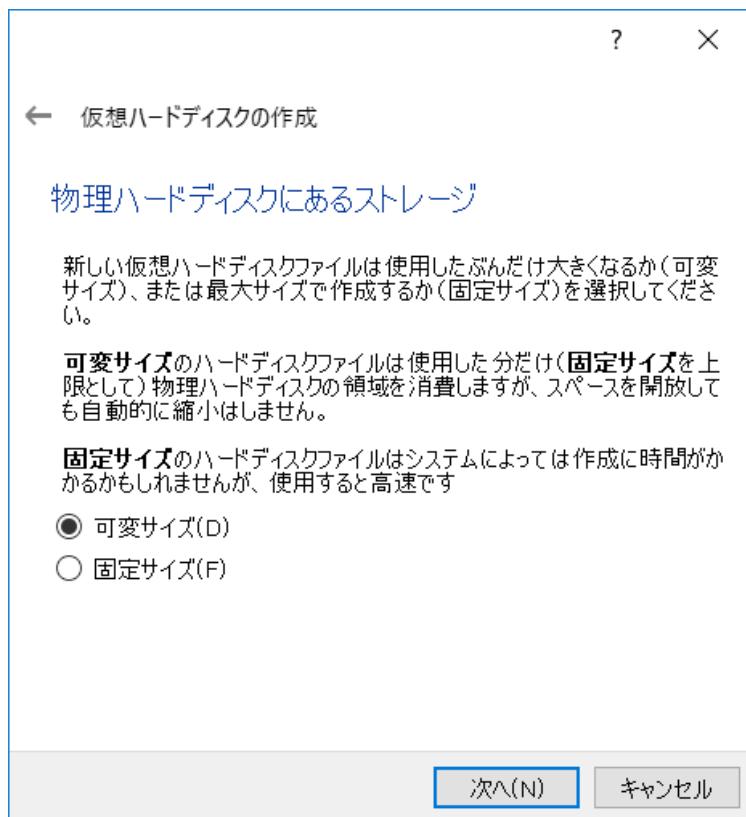
8. 「仮想ハードディスクを作成する(C)」にチェックして作成に進む。



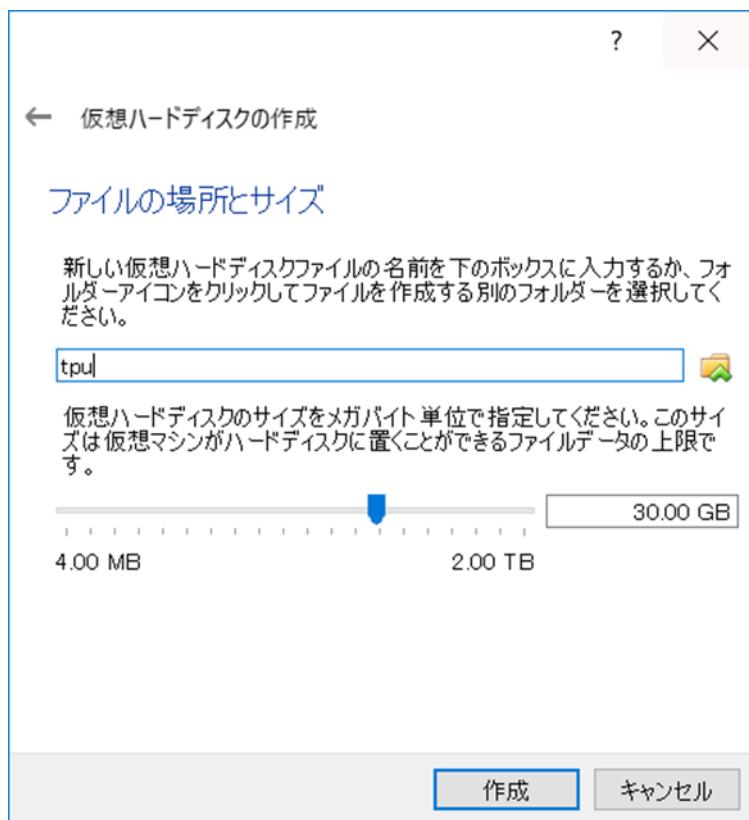
9. ハードディスクのファイルタイプを「VDI (VirtualBox Disk Image)」に選択して次に進む。



10. 物理ハードディスクにあるストレージは「可変サイズ」を選択する。



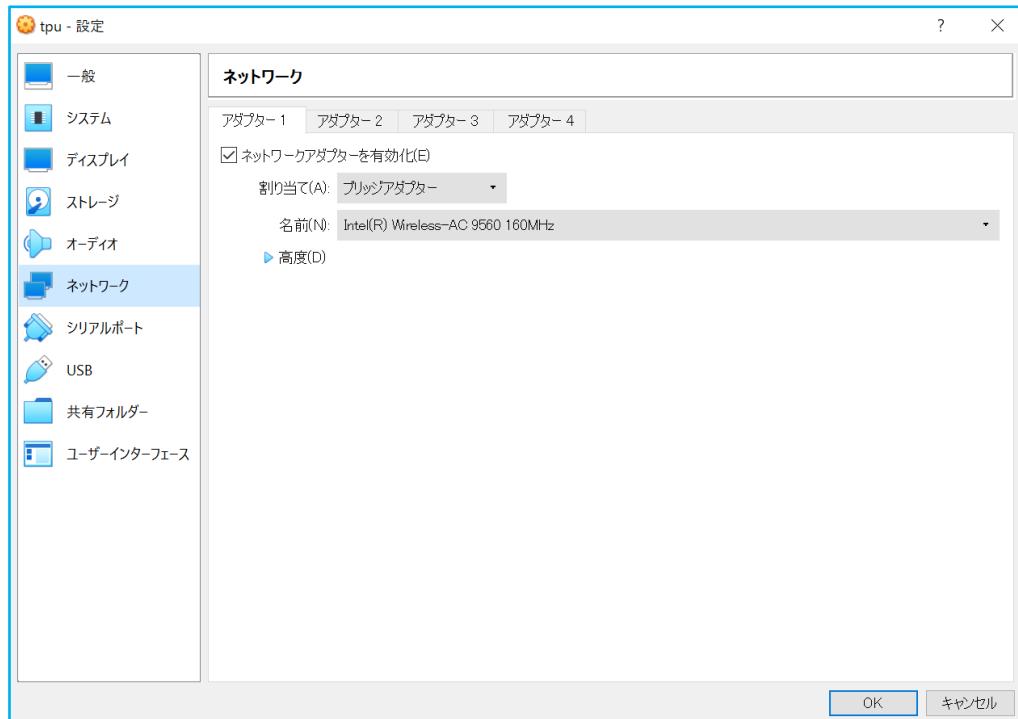
11. HDD の容量は「30GB」に設定して作成する。(必ず 30GB に増量すること)



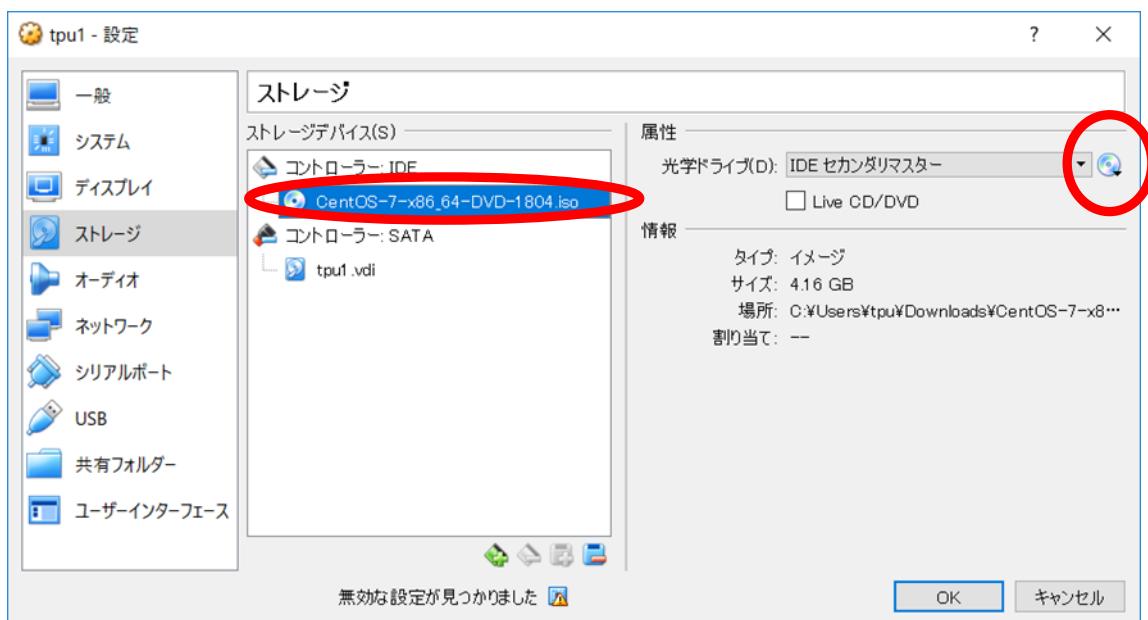
12. 作成した仮想マシンの設定を行うため、設定ボタンを押す。



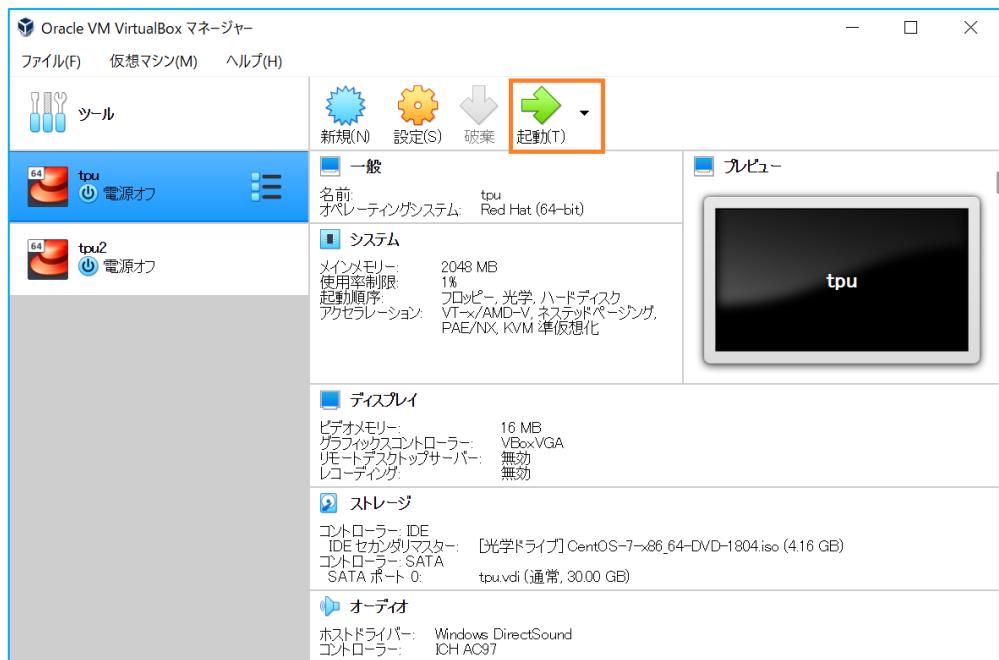
13. ネットワークからアダプター1を「ブリッジアダプター」に変更し、名前部分は無線LANのもの(この例では「Intel(R) Wireless-AC 9560 160MHz」)を選択する。
念のため、アダプター2にはNATを選択する



14. ストレージは、コントローラーIDEの「空」を選択し、右の光学ドライブ右端のディスクをクリックし、USBメモリからコピーしたファイル「CentOS-7-x86_64-DVD-1804.iso」を選択する。(ここもバージョンの関係でファイル名の数字は変わる可能性がある)

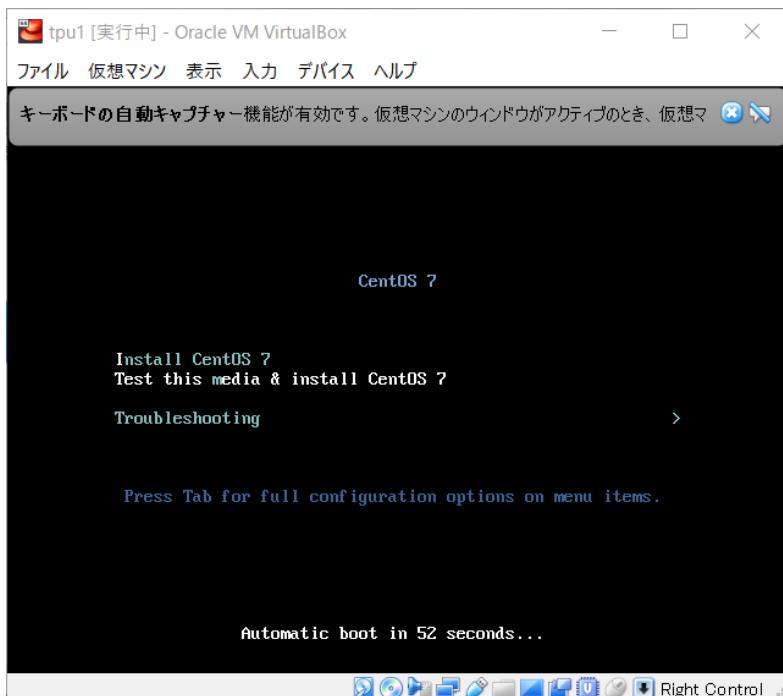


15. 起動ボタンを押し、仮想マシンを起動する。(Windows が激重になるので注意すること)

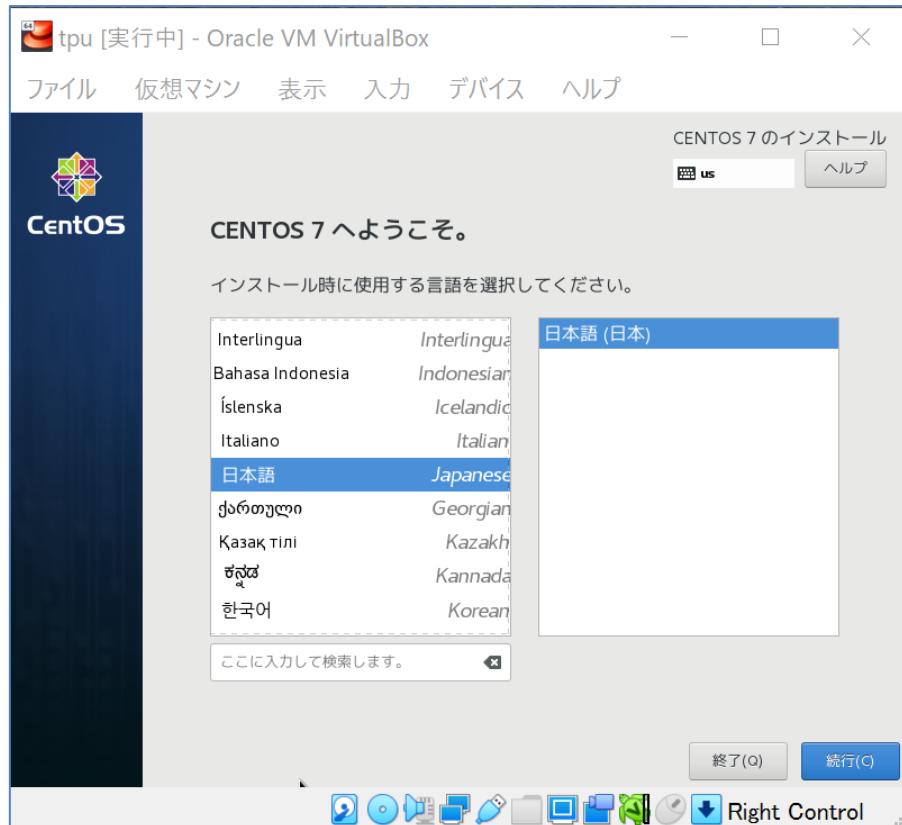


16. ここから仮想マシン上での「CentOS のインストール・セットアップ」になる。以降、仮想マシン内の操作とマシン外(Windows 側)のマウス操作で「コントロールの切り替え」が発生するので注意すること。**仮想マシン内から抜けて、Windows 側のマウスカーソルを操作するときは、右下側の「Ctrl」キーを押すことで切り替えられる。**

17. 仮想マシンの画面が起動し、CentOS 7 のインストールメニュー画面になるので、「Install CentOS7」を選択する。



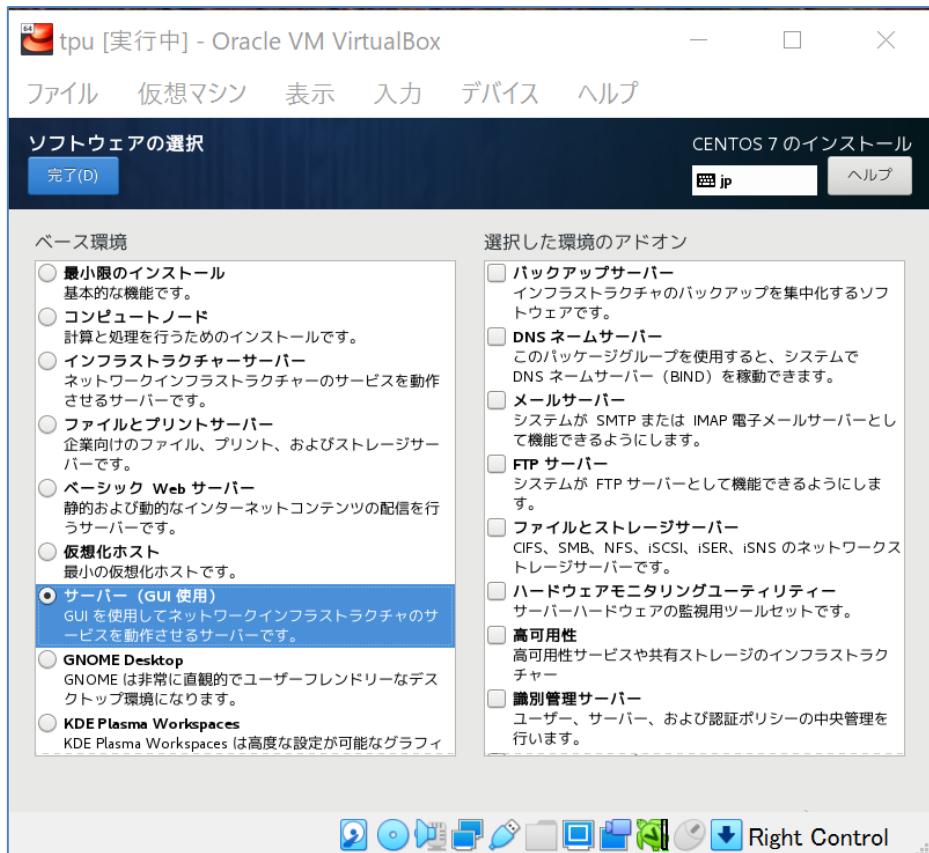
18. 使用する言語を「日本語(Japanese)」を選択し、右下の「続行」ボタンで続ける。



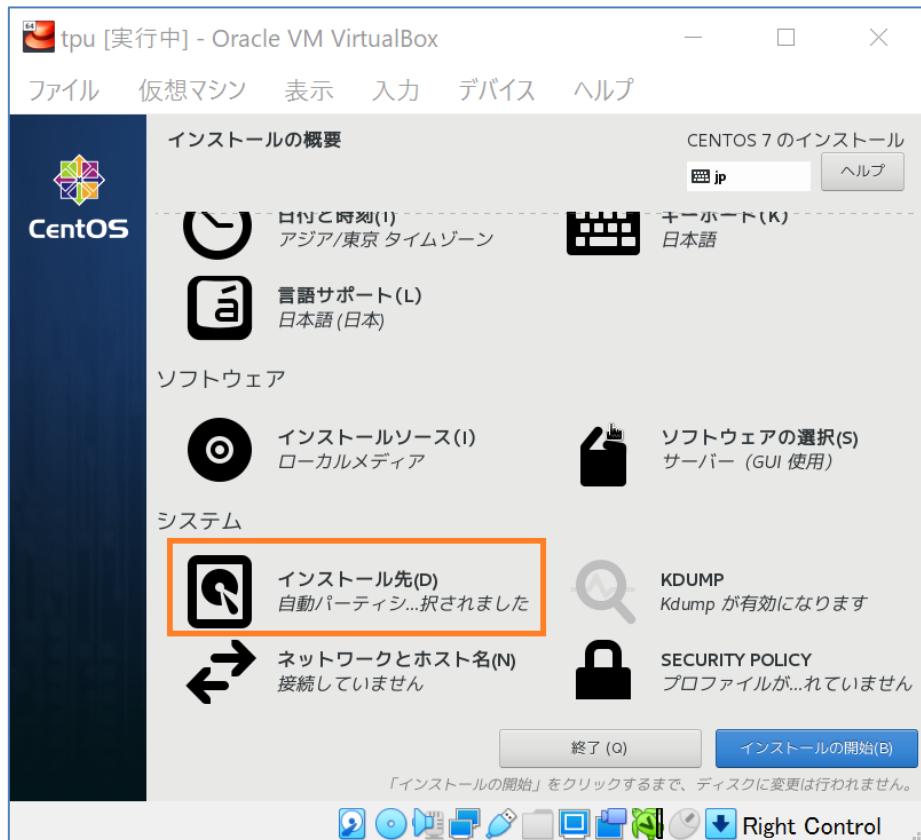
19. 「ソフトウェアの選択(S)」を選択する。



20. 「サーバー (GUI 使用)」を選択して、左上の完了で進む。



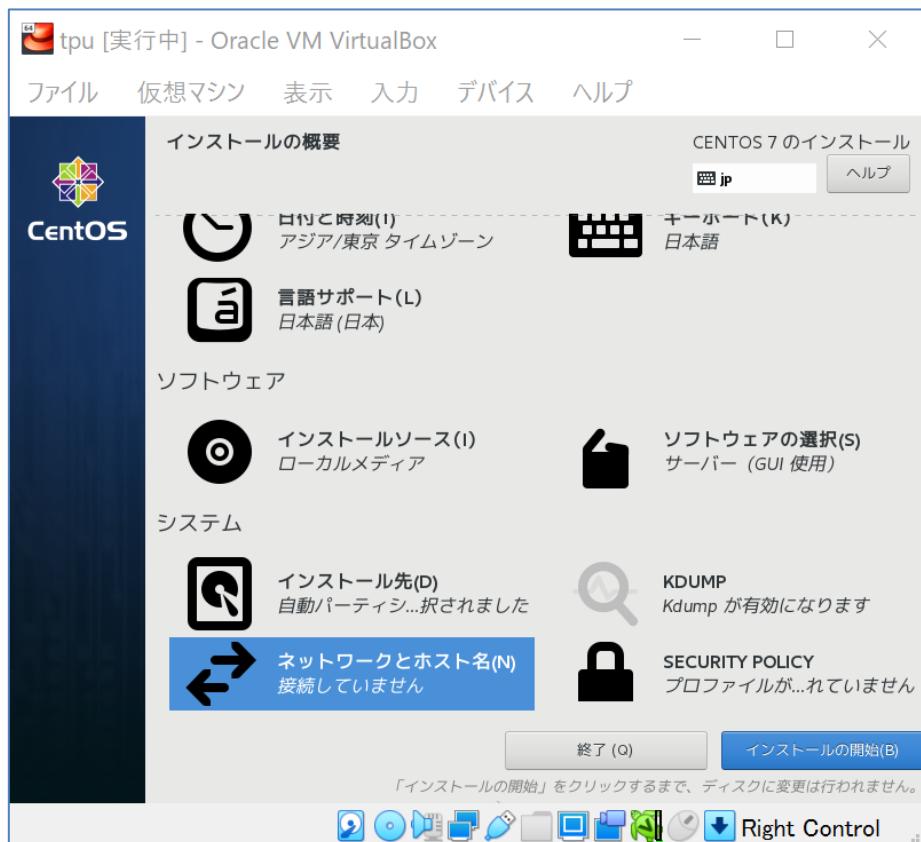
21. 「インストール先 (D)」を選択する。



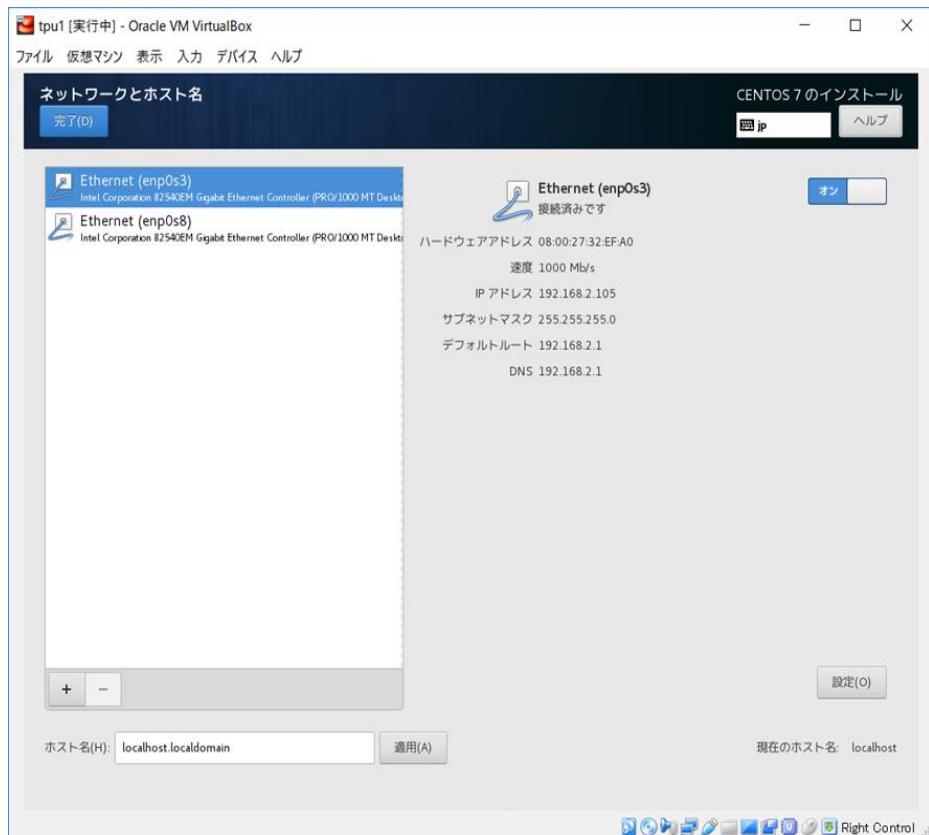
22. 既に 30GB の HDD が選択されている状態になるので、そのまま完了で進む。



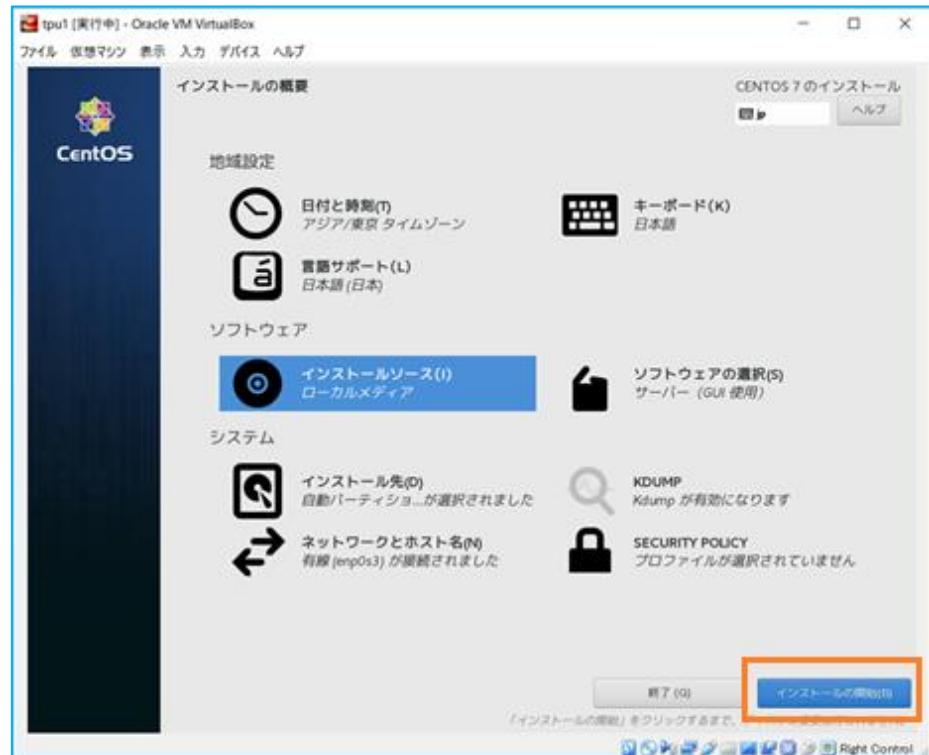
23. 「ネットワークとホスト名 (N)」を選択する



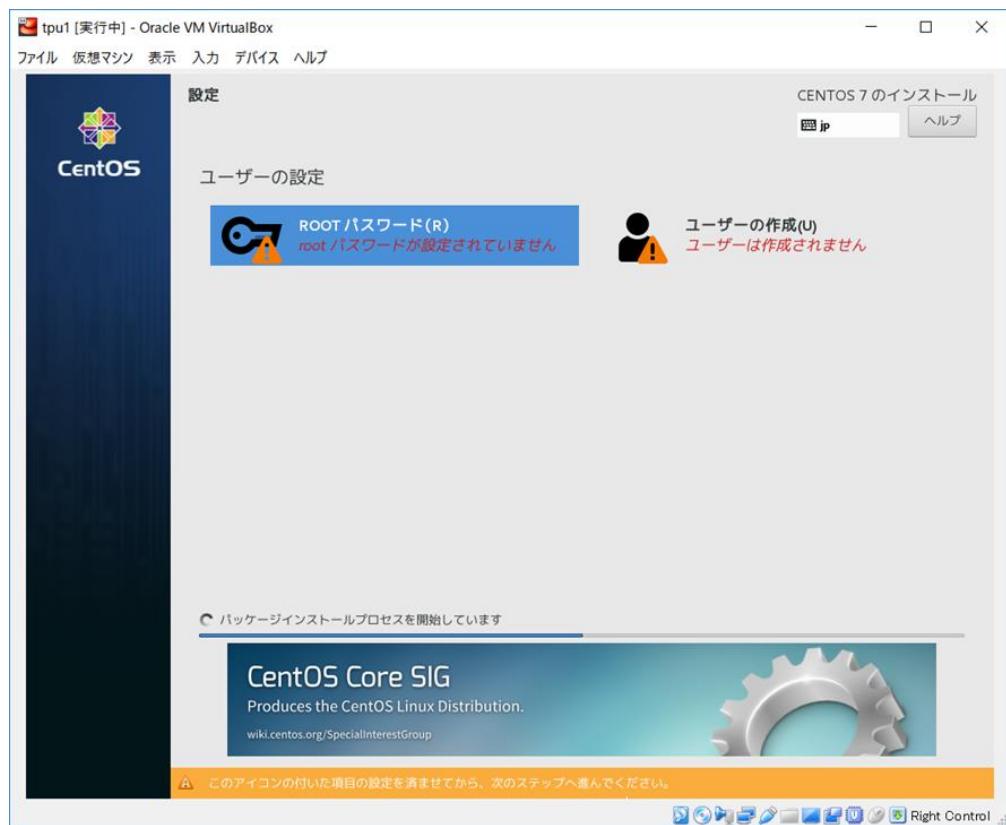
24. enp0s3 を「オン」（右上のスライドを変更）にして完了する。



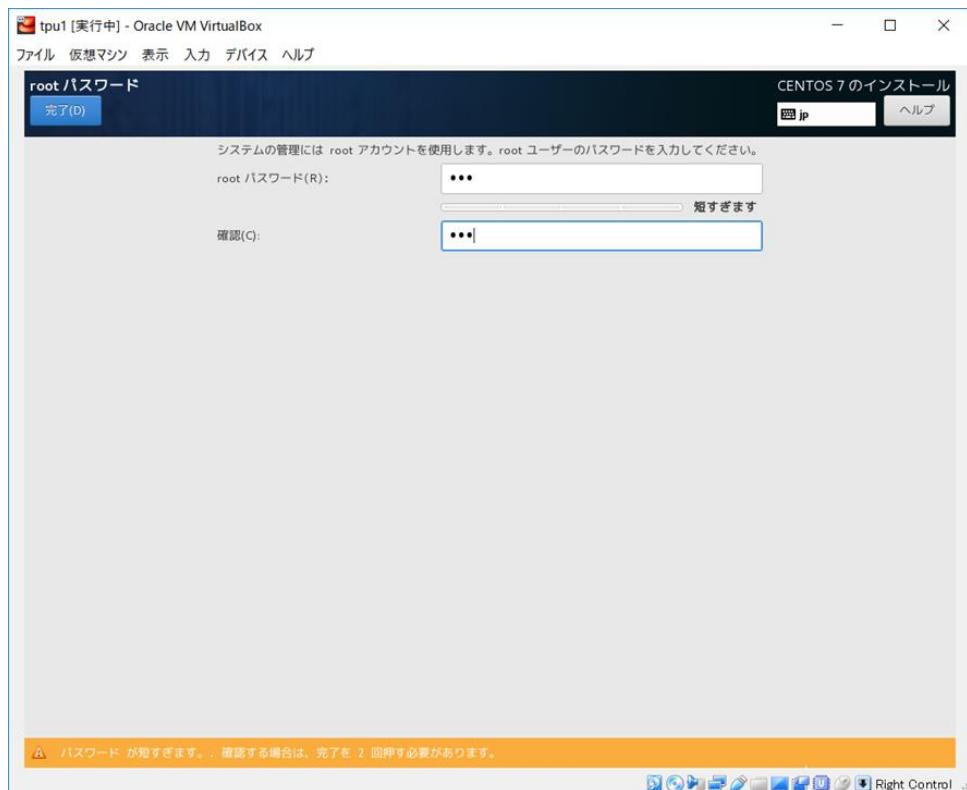
25. ここまで手順を終えたら、右下の「インストールの開始(R)」が有効になっているはずなので、これを実行する。



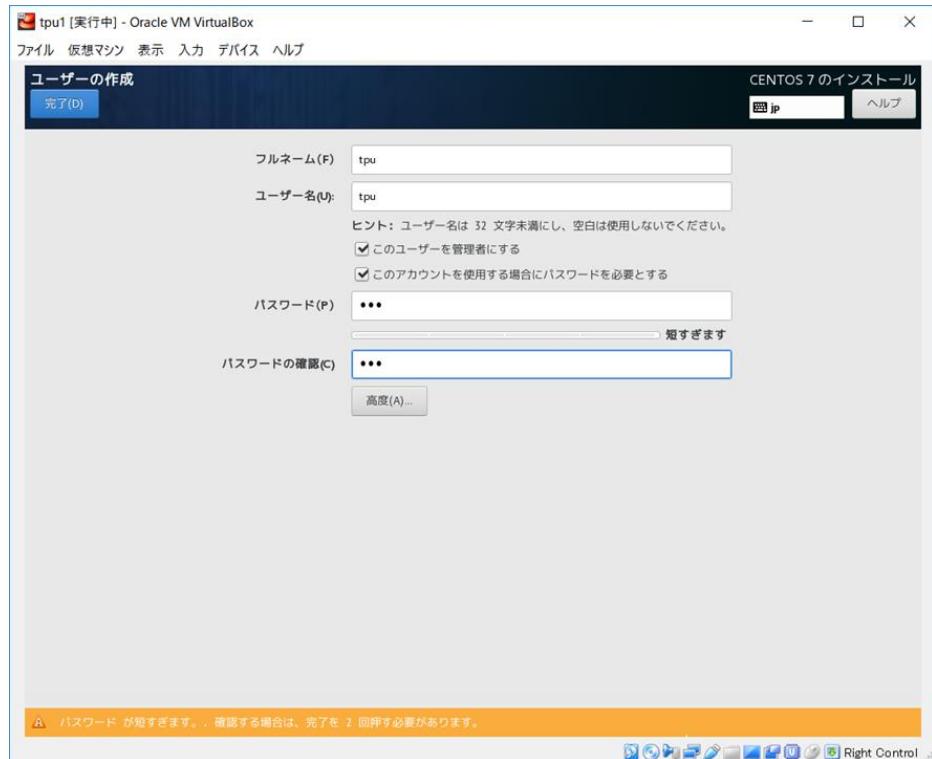
26. root パスワードの設定を行う。



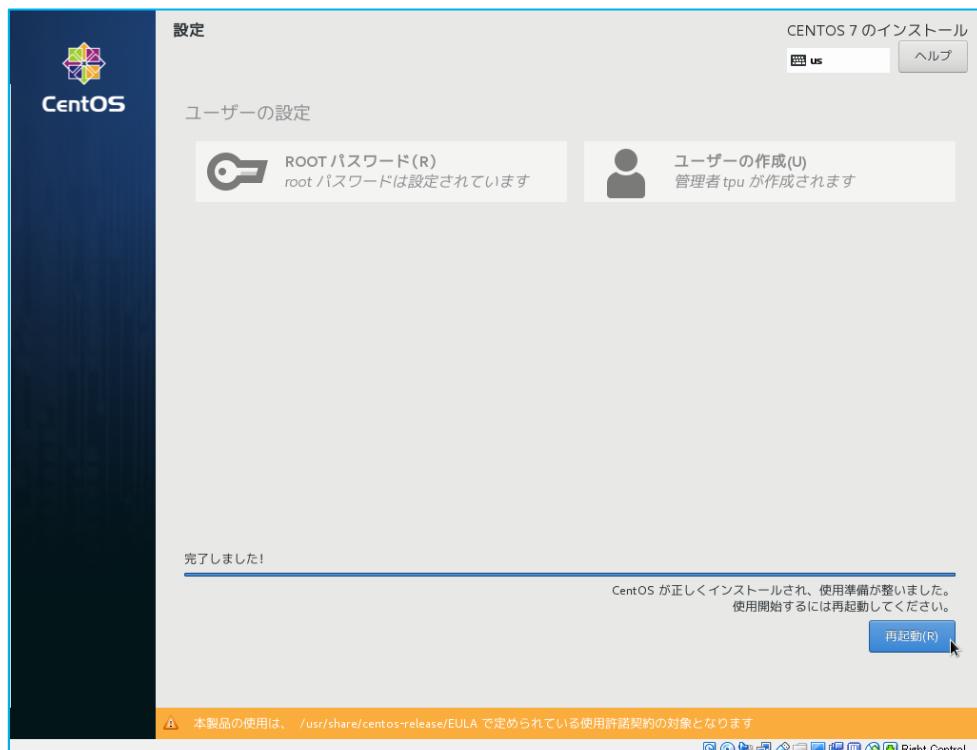
27. root パスワードは「tpu」とする。パスワードを 2 回入力して「完了」を押すと、短すぎるというエラーが表示されるが、そのまま「完了」をもう 1 回押せばそのまま進む。



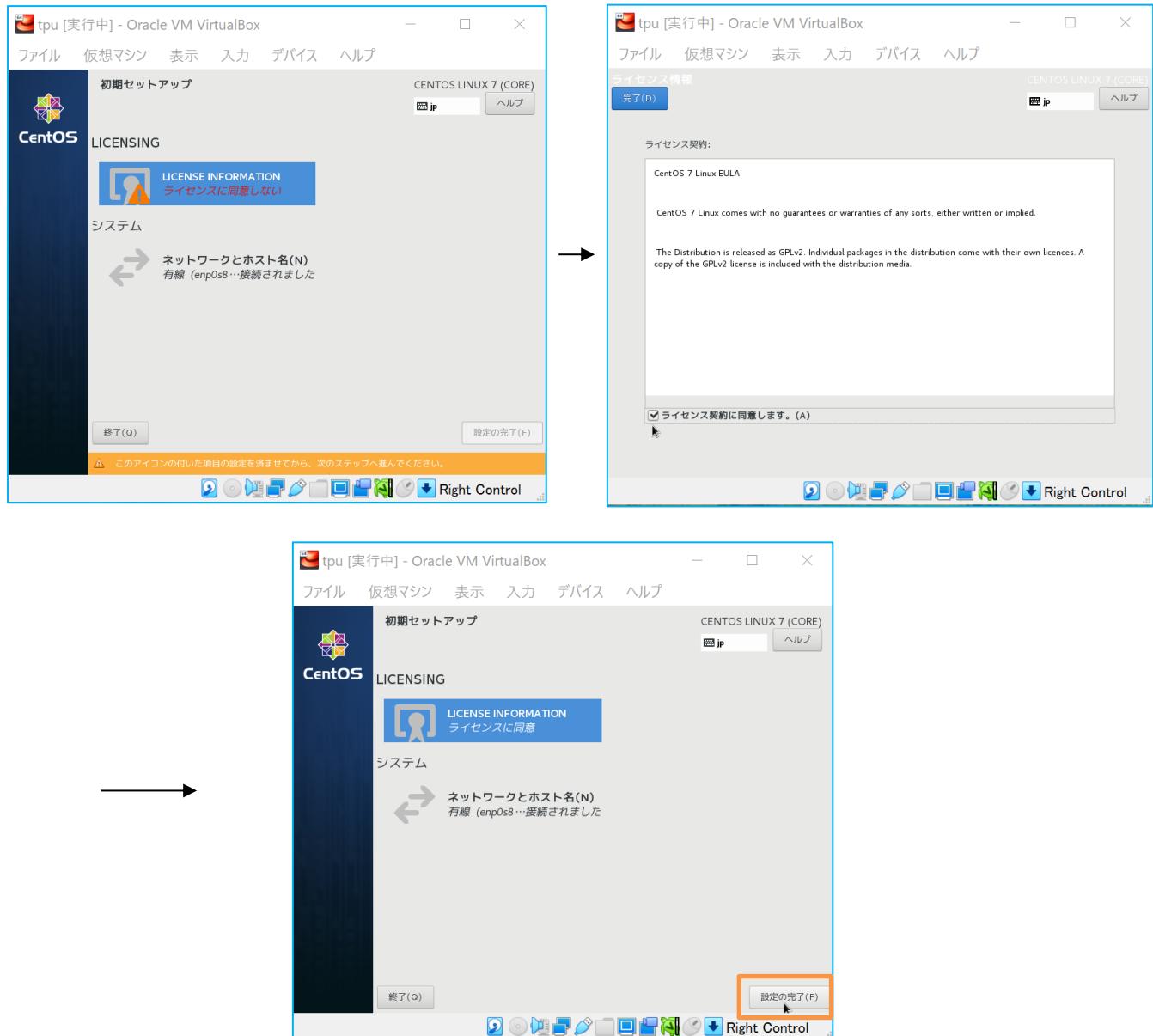
28. 「ユーザーの作成」に進み、ユーザー名を「tpu」にして、2ヶ所あるチェックを全てONにし、パスワード「tpu」を2回入力して完了を押す。こちらも同様にエラーになるが、完了をさらに押して進む。



29. インストールが完了すると、右下に「再起動」ボタンが現れるので、そのまま再起動する。
(仮想マシンの再起動なので、Windowsを再起動する必要はない)



30. CentOS 起動後、初期設定のウィンドウが開くので、ライセンスに同意し、設定を完了する。

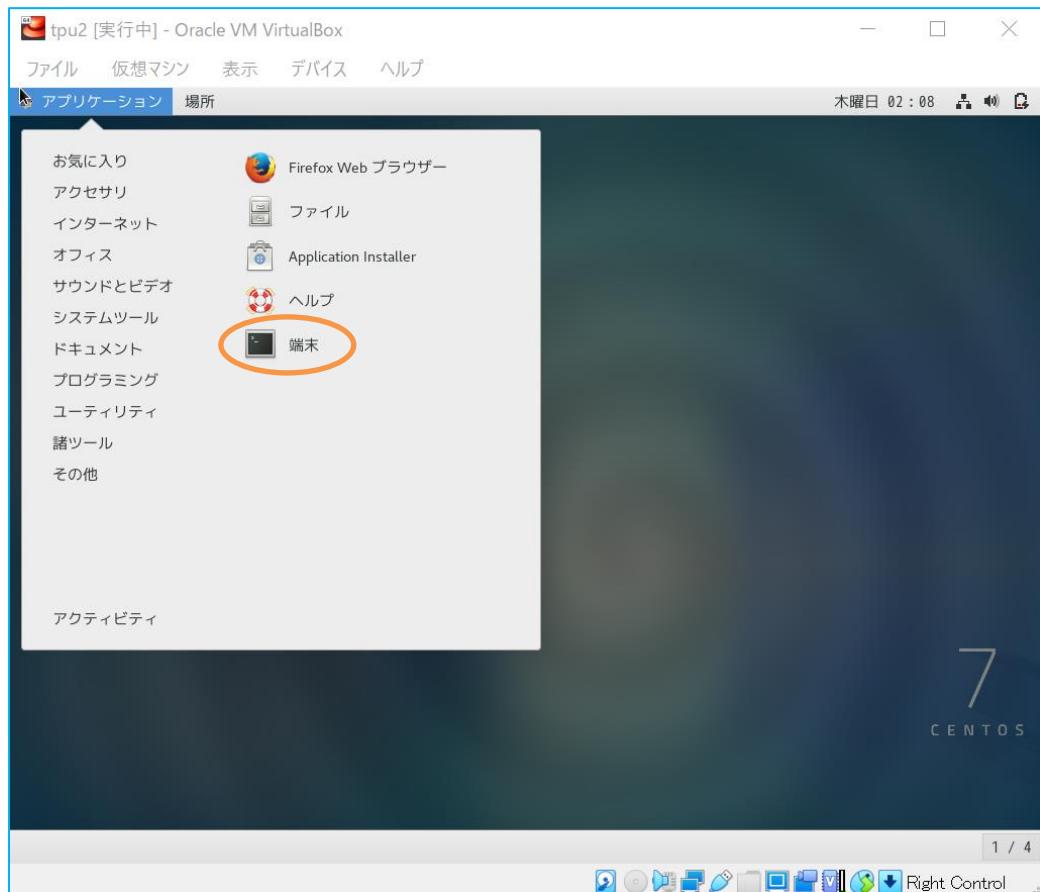


31. ログイン画面になるので「tpu」でログインする。

32. 最初のログインなので「ようこそ」ウィンドウが開くので、順に「日本語」→「日本語」→位置情報「オフ」→「スキップ」と選択し、「CentOS Linux を使い始める」をクリックする。

33. 「初めて使う方へ」のウィンドウが開くが、特に読まなくても大丈夫なので閉じる。

34. 画面左上の「アプリケーション」から、「端末」を選択して開く。以後はほぼ全てこの端末ウィンドウ内で作業を行う。以下、テキストでは端末内のコマンドラインに入力する文字列は「\$」で始まる行で示す。「\$」自体は入力する必要は無くその右からの文字列を(スペース空白を含めて)入力し、最後に「Enter」を入力する。



35. まず、sudo が正常に機能するか確認するため、以下を入力する。

```
$ sudo echo test
```

tpu のパスワード入力が要求されるので、パスワードを入力すると、「test」と表示されて終了する。これができていることを確認すること。sudo コマンドは必要な場合にはパスワードを要求してくるので、そのままパスワードを入力すること。

E-2. 以下のエラー出た時の対処

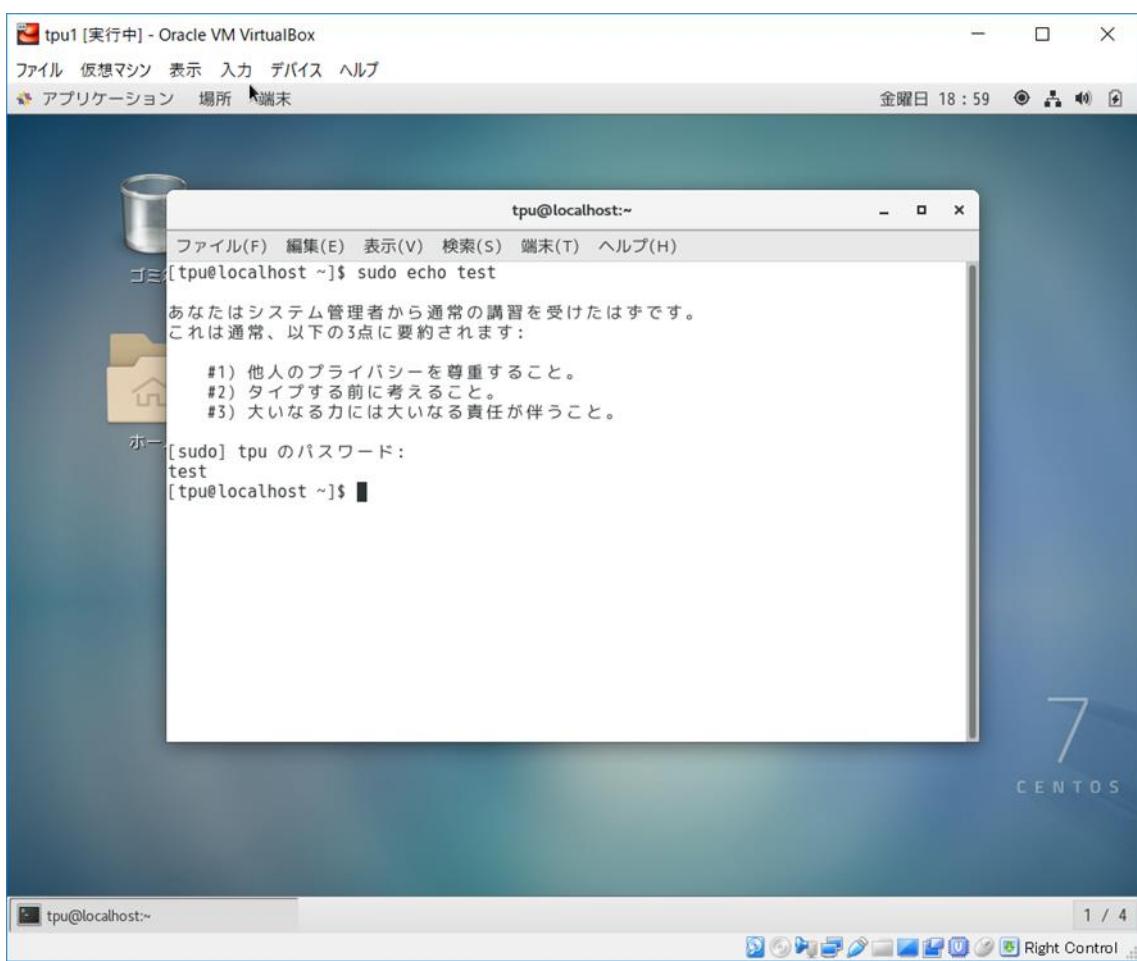
sudoers ファイル内にありません。この事象は記録・報告されます。と出たときは以下で解決

```
$ su
```

```
$ usermod -G wheel tpu
```

```
$ su tpu
```

E-2. 以下のエラー出た時の対処（終わり）



36. ここで、これから使用する vim というエディターについて説明をする。vim は主に UNIX 系の OS 上で、CUI で使われることを前提に設計された vi というエディターの改良版である。vim は、高性能なエディターであるが、いわゆる Windows のエディターとは若干操作体系が異なるので、使いこなしておおくこと。大きな違いは、「通常モード」と「挿入モード」という 2 つのモードが存在し、文字入力(追加)は基本的に「挿入モード」でしかできない点である。これを覚えておくこと。

Vim で使用する主なコマンドを以下の表に示す。全て「通常モード」の時に入力します。

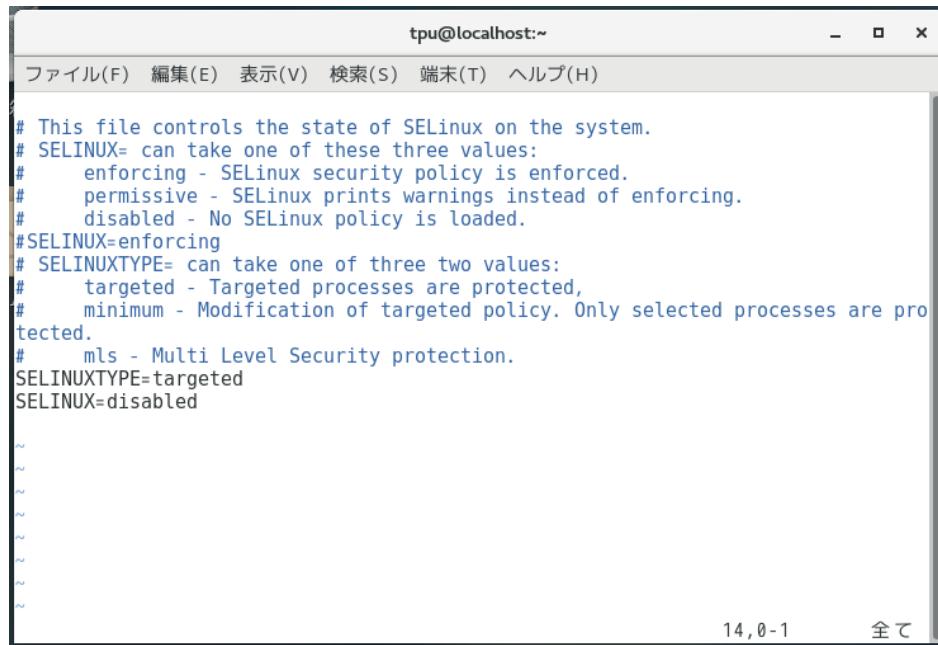
表 1 : Vim のコマンド一覧

コマンド	効果
h	カーソルを左に動かす。カーソルキー「←」でも可。
j	カーソルを下に動かす。カーソルキー「↓」でも可。
k	カーソルを上に動かす。カーソルキー「↑」でも可。
l	カーソルを右に動かす。カーソルキー「→」でも可。
x	カーソル直下の文字を消す
dd	カーソル直下の 1 行を消す
a	カーソルの場所から 1 文字進めて文字入力モードに入る
i	カーソル直下の場所から文字入力モードに入る
yy	カーソルのある 1 行をコピーする
p	上記 yy でコピーした行を貼り付ける
:wq	ファイルを保存して終了
:q	終了(ファイル変更が無い場合)
:q!	ファイルが変更されていても保存せずに強制終了

※ 「挿入モード(文字入力モード、Insert Mode と表示されることも)」から「通常モード」に戻るには、「Esc」キーを押す。

37. 早速この vim を使って、SELinux の無効化設定を行う。コマンドは以下のとおり。

```
$ sudo vim /etc/selinux/config
```



```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
#SELINUX=enforcing
# SELINUXTYPE= can take one of three two values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
SELINUX=disabled
```

上図のように、SELINUX=disabled を最後に追加し、SELINUX=enforcing の行をコメントアウト(行の

先頭に「#」を追加する。もしくは、ファイルの途中にある SELINUX=enforcing の行を直接

SELINUX=disabled に書き換える形でも良い。

ファイルを保存して vim を終了する。(コマンド「:wq」を実行)

38. さらに、手動でも SELinux を停止する。以下のコマンドを実行する。

```
$ sudo setenforce 0
```

39. SELinux の動作状態を確認する。以下のコマンドを実行する。

```
$ getenforce
```

「Permissive」もしくは「Disabled」と表示されれば OK である。ダメな場合は上の無効化手順をもう一度実行してみること。

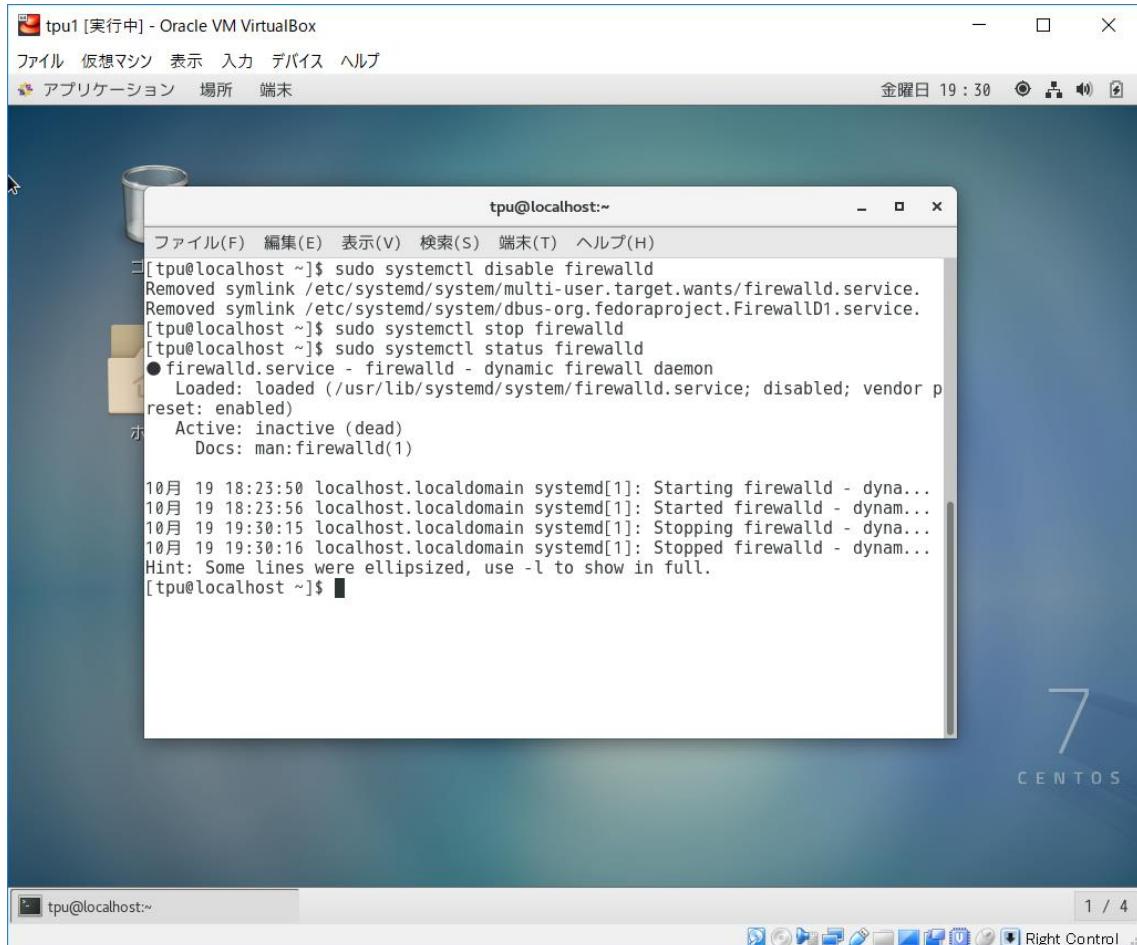
40. 次に、仮想マシンのファイアウォールを無効化する。以下のコマンドを実行する。

```
$ sudo systemctl disable firewalld  
$ sudo systemctl stop firewalld
```

41. ファイアウォールが停止したかどうかの確認は以下のコマンドができる。

```
$ systemctl status firewalld
```

全部英語で読みづらいが、「Active: inactive (dead)」の表示があればOKである。



42. 無線機の人は、ここで、一度仮想マシンの操作から自分のPCの操作に

切り替えて、Wi-Fi「Buffalo-A-EA60」に接続する。

→その後、仮想マシンの操作に戻り、画面右上のネットワークが
「enp0s3」接続になっているか確認する。

T-2. Wi-Fiのパスワード

パスワードは「t7vt35jbct8b7」であることを伝える。

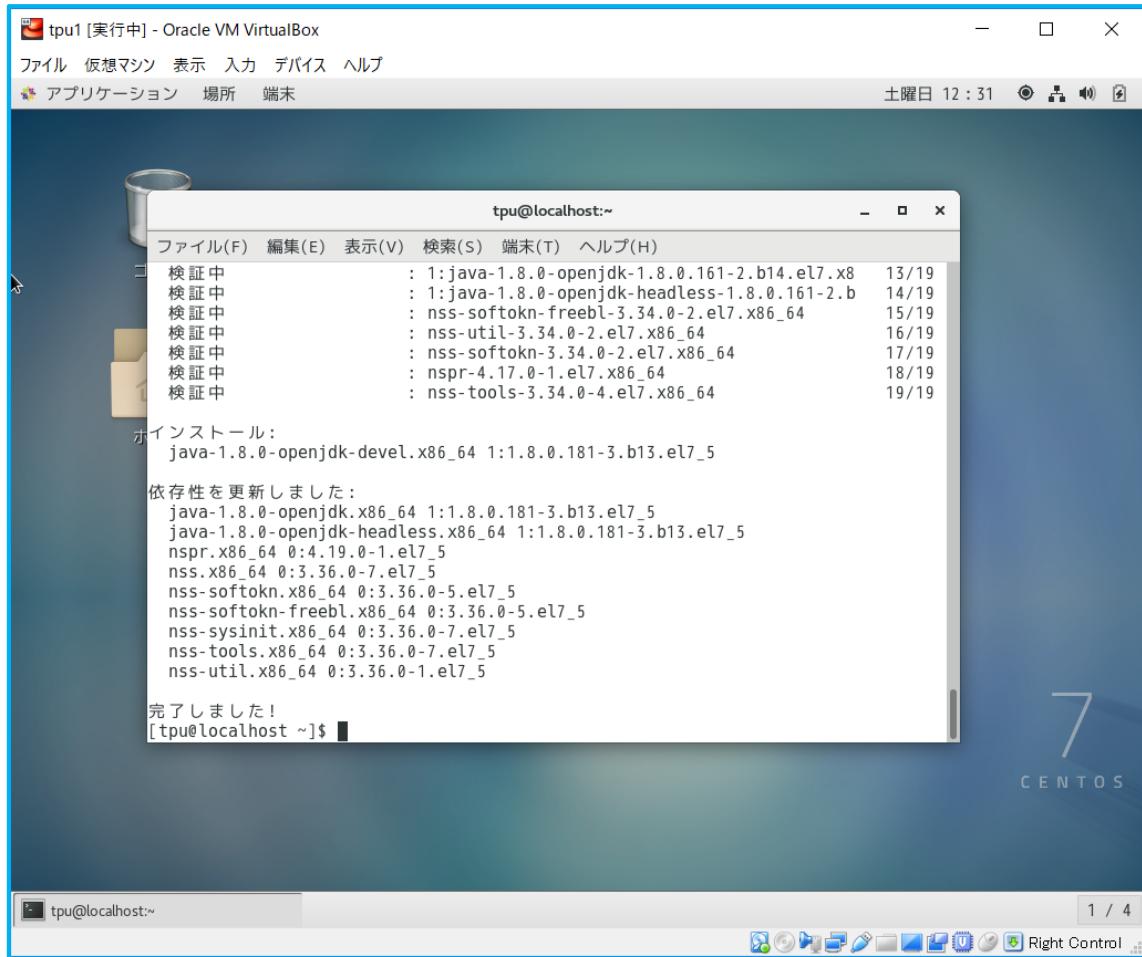
T-2. Wi-Fiのパスワード(終わり)



43. 次に、Java JDK および Mpich のインストールをするために、以下のコマンドを入力する。

```
$ sudo yum -y install java-1.8.0-openjdk-devel mpich mpich-devel gcc-c++ openssl-devel
```

下のような画面になれば成功している。(Hadoop の実験だけ行うなら飛ばしてもよい)



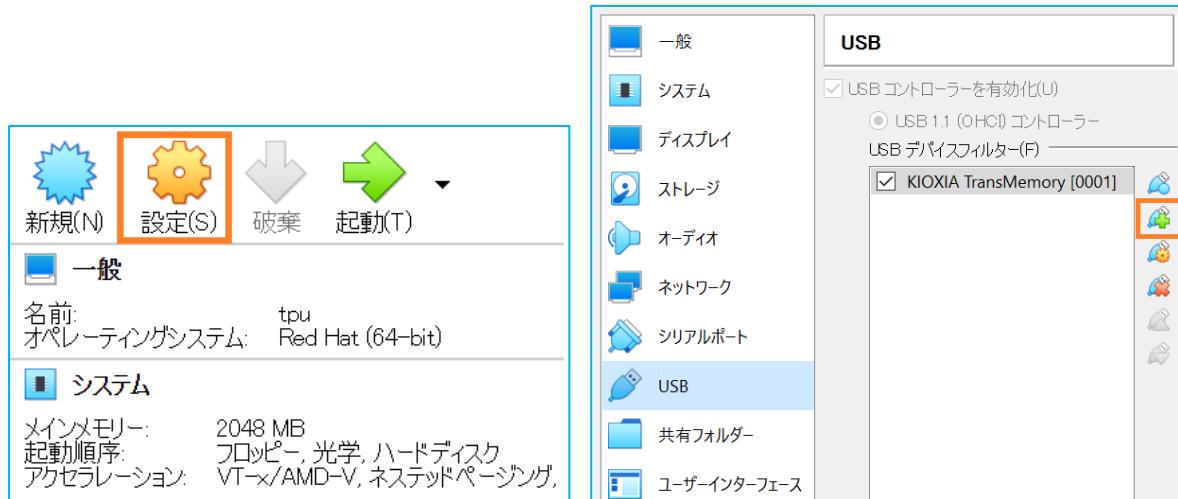
44. 次に、hadoop ソフトウェア本体のダウンロードするために、以下のコマンドを順番に入力する。

```
$ cd
```

```
$ wget https://archive.apache.org/dist/hadoop/common/hadoop-2.8.5/hadoop-2.8.5.tar.gz
```

※ダウンロードの残り時間が異常に長い(数時間など)場合、以下の手順でダウンロード

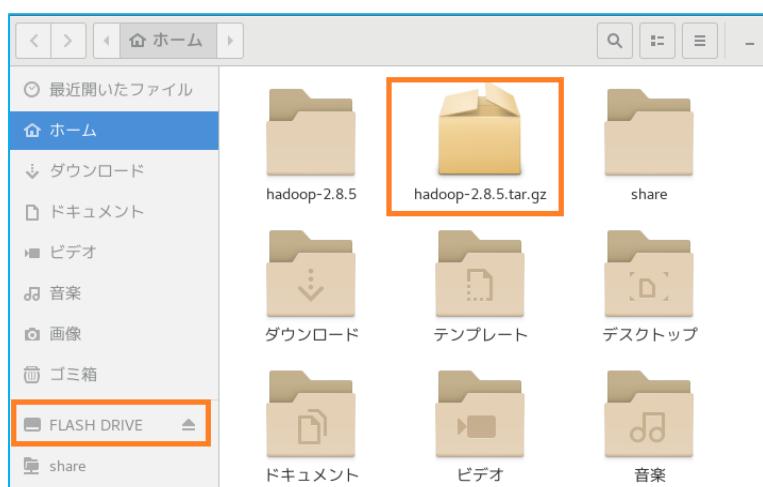
- 1) 仮想マシンウィンドウを×で閉じ、「仮想マシンの電源 OFF」を選択して OK する
- 2) Virtual Box マネージャの「設定」から USB の項目へ行く



3) 「hadoop-2.8.5.tar.gz」が入った USB を配布するので、それを PC に挿した状態で画面右側の「+」から「KIOXIA」を選択し、チェックを入れる

4) 一度 USB を取り外し、仮想マシンを起動する

5) 起動後 USB を再び挿し、仮想マシンデスクトップ上の「ホーム」フォルダを開く



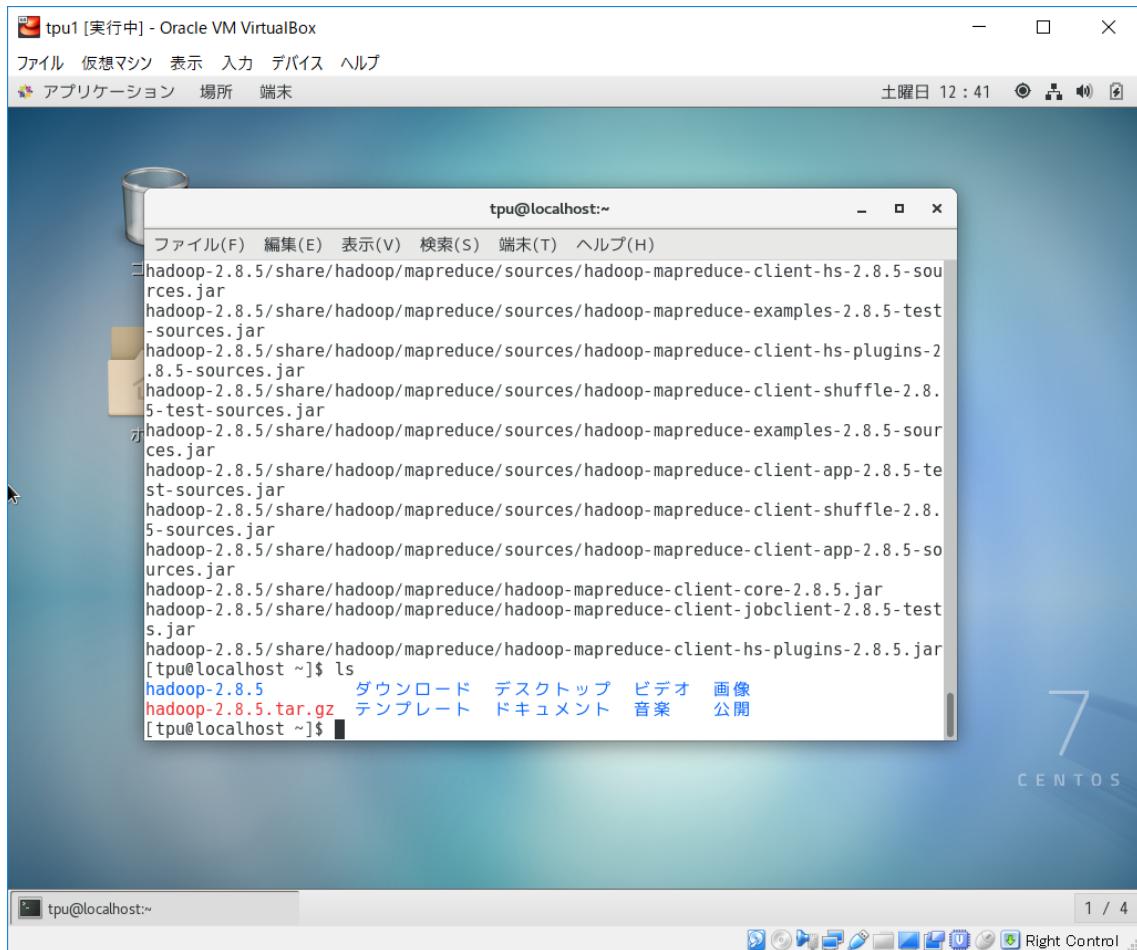
6) 認識されていれば、左側に「KIOXIA」が表示されるので、USB 内の「hadoop-2.8.5.tar.gz」を「ホーム」にドラッグしてコピペする。

45. その後、ホームディレクトリに hadoop-2.8.5 を展開したいので以下のコマンドを入力する。

```
$ tar xvzf hadoop-2.8.5.tar.gz
```

46. ホームディレクトリで ls と入力して下図のように青色の「hadoop-2.8.5」と赤色の「hadoop-2.8.5.tar.gz」が両方とも表示されれば、成功している。

```
$ ls
```



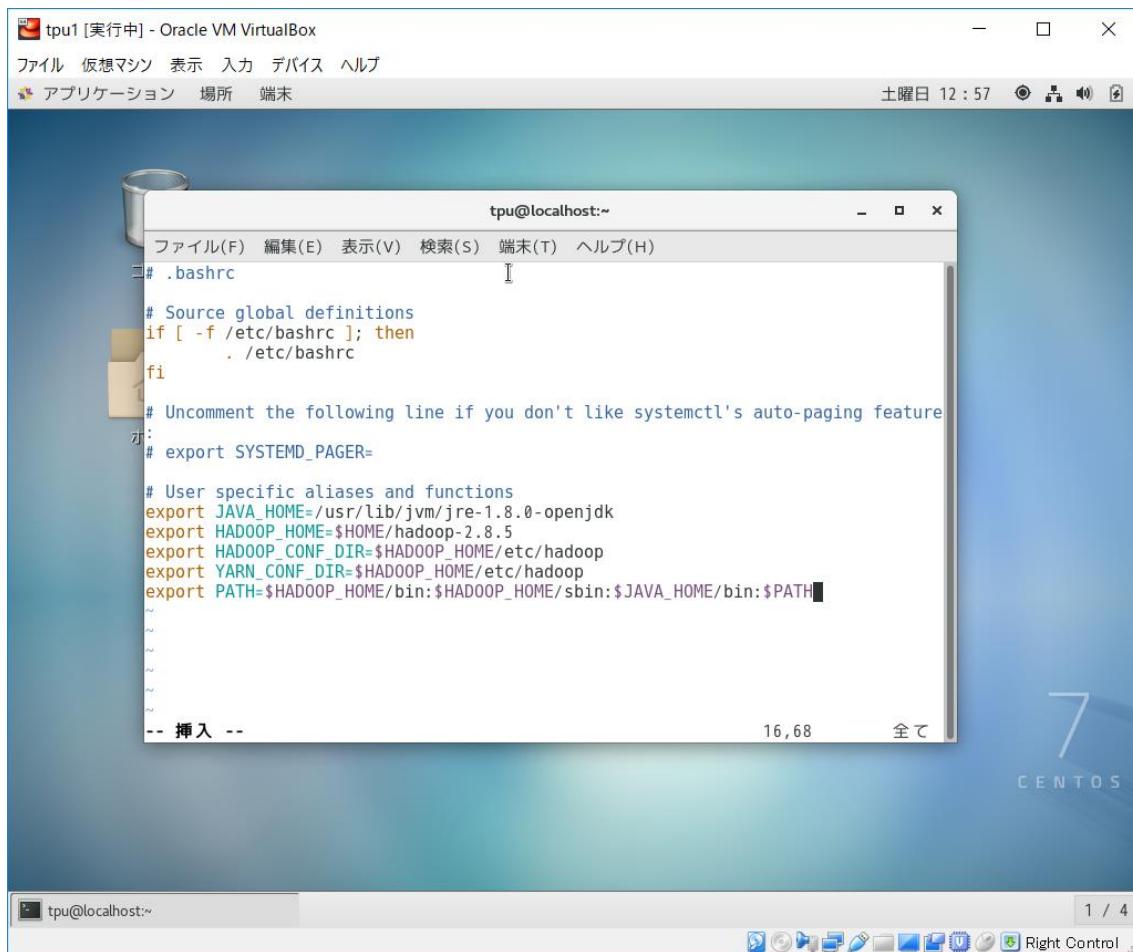
47. .bashrc ファイルを変更し、bash 環境の設定を変更する。以下のコマンドで編集を開始する。

```
$ vim .bashrc
```

ファイルの最後に以下の 7 行を追加しファイルを保存して終了する。

(テキストから VirtualBox 内にはコマンドをコピペできないので、根気よくやること...)

```
$ export JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk
$ export HADOOP_HOME=$HOME/hadoop-2.8.5
$ export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
$ export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
$ export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$JAVA_HOME/bin:$PATH
$ export PATH=/usr/lib64/mpich/bin:$PATH
$ export HADOOP_CLASSPATH=/usr/lib/jvm/java-1.8.0-openjdk/lib/tools.jar
```



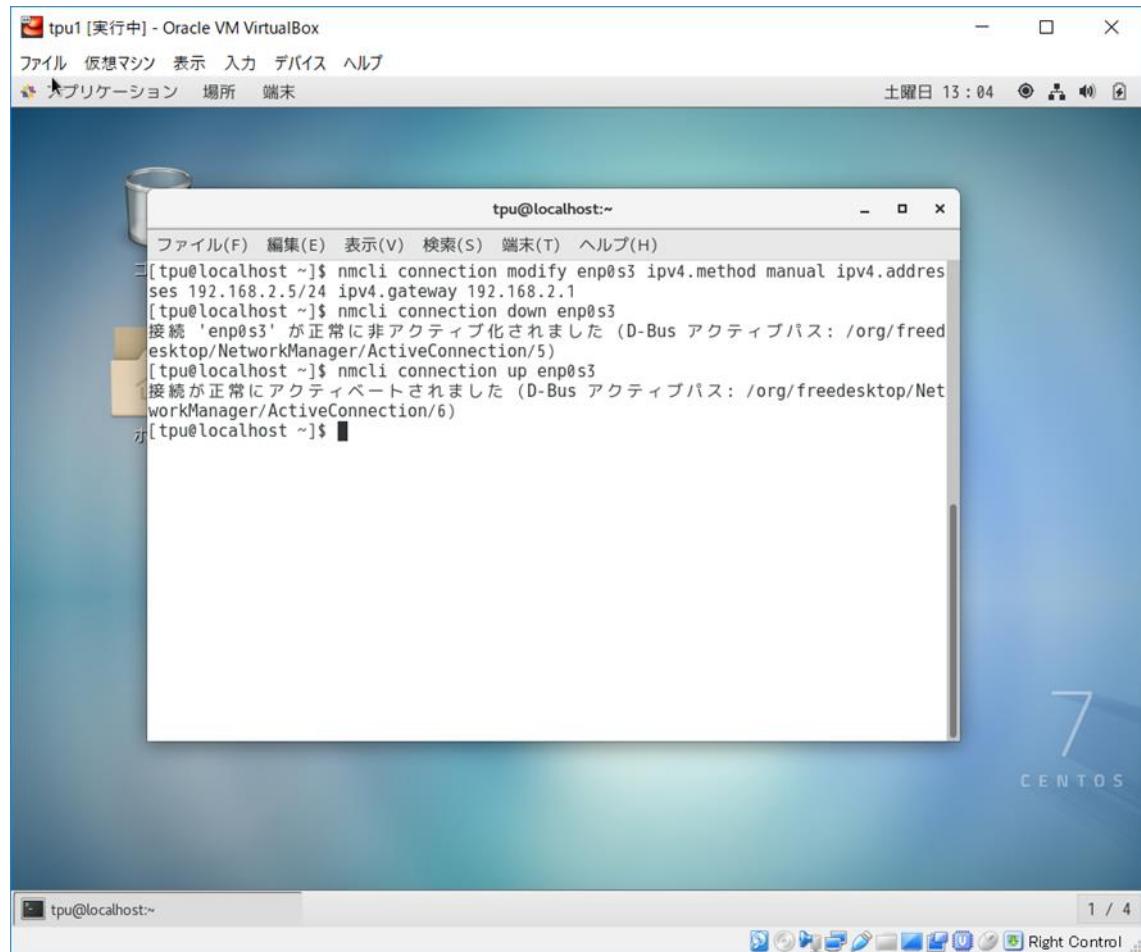
48. .bashrc ファイルの変更を反映するために以下のコマンドを入力する。

```
$ source .bashrc
```

49. 自分の仮想マシン自身の IP アドレスを固定番号に設定する。以下のコマンドを順番に入力する。
XXX の部分は、自分の学籍番号下 3 術部分の数字に置き換える。例えば、1715010 の場合は、XXX の部分は 10 となる。(過年度生は番号重複の可能性があるため TA に相談すること)

```
$ nmcli connection modify enp0s3 ipv4.method manual ipv4.addresses 192.168.11.XXX/24
$ nmcli connection modify enp0s3 ipv4.gateway 192.168.11.1
$ nmcli connection down enp0s3
$ nmcli connection up enp0s3
```

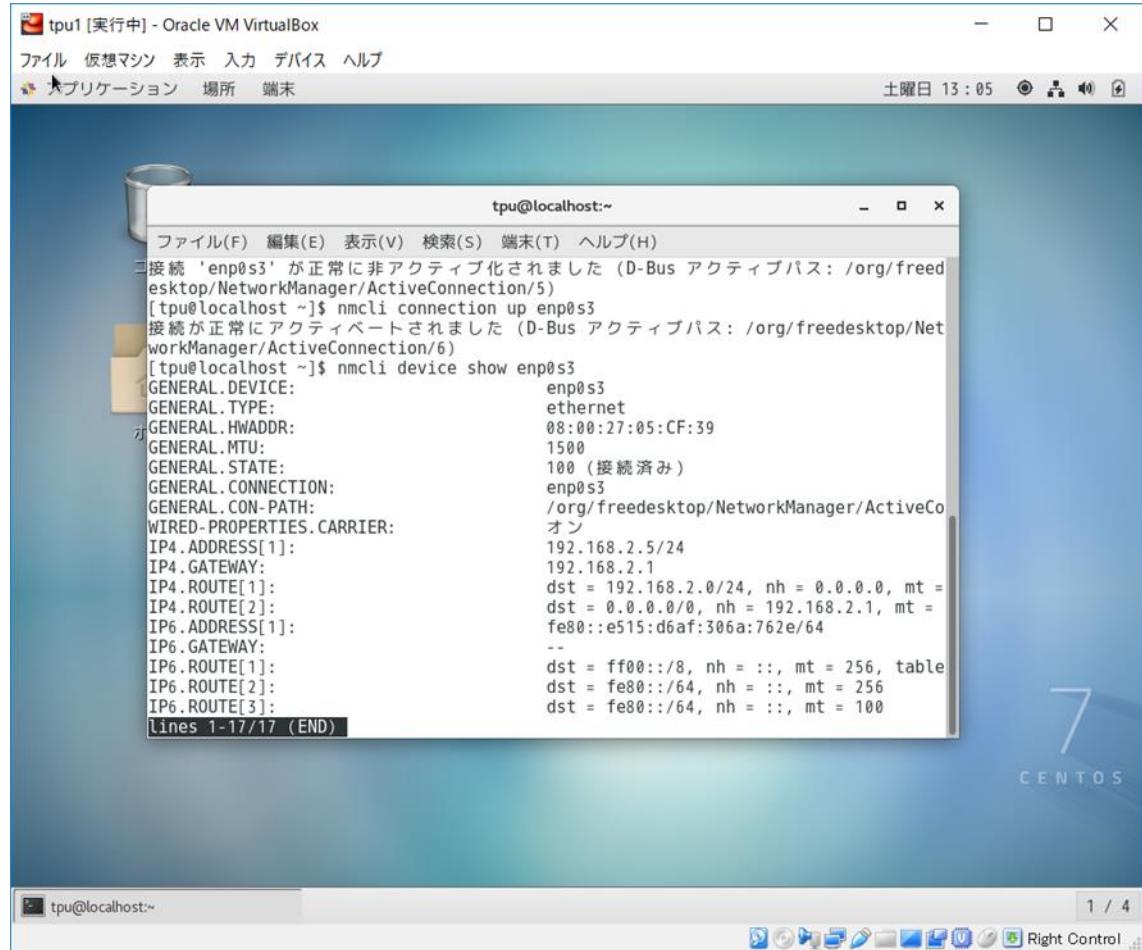
下図は、学籍番号 1715005 の場合の画面である。



50. 以下のコマンドを入力して、自分の設定した IP 番号で動作しているか確認できればよい。
(GENERAL.STATE の「接続済み」表示など)

```
$ nmcli device show enp0s3
```

確認後 Ctrl+C で抜ける。



```
tpu1 [実行中] - Oracle VM VirtualBox
ファイル 仮想マシン 表示 入力 デバイス ヘルプ
アプリケーション 場所 端末
土曜日 13:05

tpu@localhost:~
```

ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)

```
接続 'enp0s3' が正常に非アクティブ化されました (D-Bus アクティブパス: /org/freedesktop/NetworkManager/ActiveConnection/5)
[tpu@localhost ~]$ nmcli connection up enp0s3
接続が正常にアクティベートされました (D-Bus アクティブパス: /org/freedesktop/NetworkManager/ActiveConnection/6)
[tpu@localhost ~]$ nmcli device show enp0s3
GENERAL.DEVICE: enp0s3
GENERAL.TYPE: ethernet
GENERAL.HWADDR: 08:00:27:05:CF:39
GENERAL.MTU: 1500
GENERAL.STATE: 100 (接続済み)
GENERAL.CONNECTION: enp0s3
GENERAL.CON-PATH: /org/freedesktop/NetworkManager/ActiveConnection/6
WIRED-PROPERTIES.CARRIER: オン
IP4.ADDRESS[1]: 192.168.2.5/24
IP4.GATEWAY: 192.168.2.1
IP4.ROUTE[1]: dst = 192.168.2.0/24, nh = 0.0.0.0, mt =
IP4.ROUTE[2]: dst = 0.0.0.0/0, nh = 192.168.2.1, mt =
IP6.ADDRESS[1]: fe80::e515:d6af:306a:762e/64
IP6.GATEWAY: --
IP6.ROUTE[1]: dst = ff00::/8, nh = ::, mt = 256, table =
IP6.ROUTE[2]: dst = fe80::/64, nh = ::, mt = 256
IP6.ROUTE[3]: dst = fe80::/64, nh = ::, mt = 100
lines 1-17/17 (END)
```

7 CENTOS

```
tpu@localhost:~ 1 / 4
```

Right Control

51. マスターPCに通信できるか確かめる。

```
$ ping 192.168.11.200(master0 の場合)
```

```
$ ping 192.168.11.201(master1 の場合)
```

```
$ ping 192.168.11.202(master2 の場合)
```

```
$ ping 192.168.11.203(master3 の場合)
```

以下のように表示されていれば通信できている(master1 の場合)。

```
PING 192.168.11.201 (192.168.11.201) 56 (84) bytes of data.
```

```
64 bytes from 192.168.11.201: icmp_seq=1 ttl=64 time=0.111 ms
```

```
64 bytes from 192.168.11.201: icmp_seq=2 ttl=64 time=0.156 ms  . . .
```

通信できていない場合以下を確認してみる

- LANケーブルが接続されているか確認する。
- 仮想マシンの右上の電池のマークのところをクリックして「Ethernet(enp0s3)」が接続状態になっていることを確認する。
- 仮想マシンの右上の電池のマークのところをクリックして右下の電源ボタンから電源オフを選択し、もう一度起動する。
- マスターPCの方でも上記3つを実行する。
- これでできなかった場合はTAを呼ぶ。

52. 仮想マシンのホスト名を設定する。以下のコマンドを順番に入力する。XXXの部分は、先ほどと同じく学籍番号下2(3)桁数字にする。

```
$ sudo hostname slaveXXX
```

```
$ echo "slaveXXX" | sudo tee /etc/hostname > /dev/null
```

以下のコマンドを入力して「slaveXXX」と表示されれば成功している。

```
$ hostname
```

下図は、学籍番号1715005の例である。



53. /etc/hosts ファイルの編集をする。このファイルは、IP 番号とホスト名の対応表である。

以下のコマンドで編集を開始する。

```
$ sudo vim /etc/hosts
```

既にある 127.0.0.1 と::1 で始まる 2 行はそのままに、その後ろに以下の行を追加する。

XX は master の IP アドレス、X は master 番号、○は自分の学籍番号下 2 桁、□以降は班の他の人の学籍番号下 2 桁を入力していく、実験をやる組全員分の IP 番号と slave 番号を設定する。

192.168.11.XX masterX

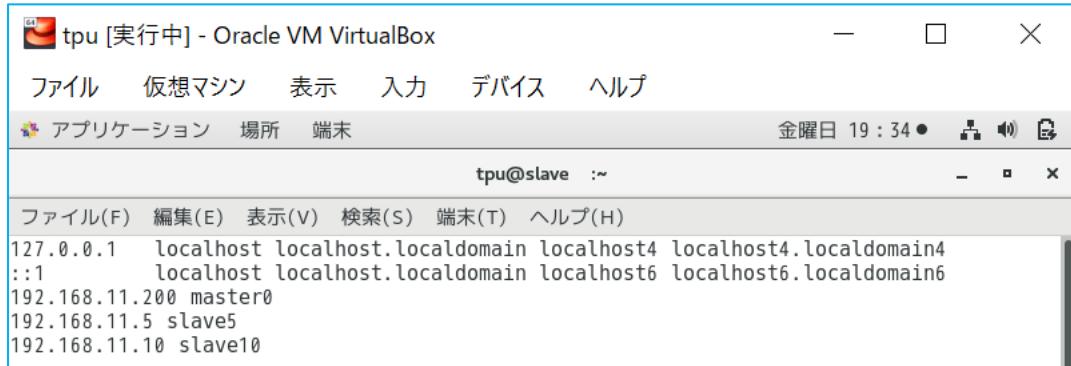
192.168.11.○ slave○

192.168.11.□ slave□

•
•
•

192.168.11.△ slave△

下図は例である。



```
tpu [実行中] - Oracle VM VirtualBox
tpu@slave:~$ cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.11.200 master0
192.168.11.5 slave5
192.168.11.10 slave10
```

54. data ディレクトリを作成する。以下のコマンドを実行する。

```
$ mkdir $HADOOP_HOME/data
```

ディレクトリができたか確認するには、以下のコマンドを入力し、下図のようになればよい。

```
$ ls $HADOOP_HOME
```



A screenshot of a terminal window titled 'tpu1 [実行中] - Oracle VM VirtualBox'. The window shows the following command history:

```
tpu@localhost:~$ mkdir $HADOOP_HOME/data
[tpu@localhost ~]$ ls $HADOOP_HOME
LICENSE.txt README.txt data include libexec share
NOTICE.txt bin etc lib sbin
[tpu@localhost ~]$
```

55. Java の動作確認をする。以下のコマンドを入力し、下図のようになればよい。

```
$ java -version
```



A screenshot of a terminal window titled 'tpu1 [実行中] - Oracle VM VirtualBox'. The window shows the following command history:

```
tpu@localhost:~$ java -version
openjdk version "1.8.0_181"
OpenJDK Runtime Environment (build 1.8.0_181-b13)
OpenJDK 64-Bit Server VM (build 25.181-b13, mixed mode)
[tpu@localhost ~]$
```

56. 同様に、hadoop の動作確認をする。以下のコマンドを入力し、下図のようになればよい。

```
$ hadoop version
```



A screenshot of a terminal window titled 'tpu1 [実行中] - Oracle VM VirtualBox'. The window shows the following command history:

```
tpu@localhost:~$ hadoop version
Hadoop 2.8.5
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r 0b8464d75227fce
e2c6e7f2410377b3d53d3d5f8
Compiled by jdu on 2018-09-10T03:32Z
Compiled with protoc 2.5.0
From source with checksum 9942ca5c745417c14e318835f420733
This command was run using /home/tpu/hadoop-2.8.5/share/hadoop/common/hadoop-com
mon-2.8.5.jar
[tpu@localhost ~]$
```

57. ssh の設定を TA にしてもらう。

T-3. ssh の設定

-----ここは絶対ミスるな.SSH の設定-----

t-1. ssh-key の配布 (master 側で入力)

```
$ sshpass -p "tpu" ssh-copy-id -i ~/.ssh/id_rsa.pub -o "StrictHostKeyChecking no" slaveXXX
```

※slave の台数分実行

(2行目の前にマスターの方でも手順 53.を行い、マスター自身と、公開鍵を配布する slave の IP 番号・
ホスト名を記載しておく)

-----ここは絶対ミスるな-----

——動かないとき——

t-2. \$ ssh-add ~/.ssh/id_rsa

——動かないとき終わり——

t-3. ※master 自身と、各 slave にパスワードなしで ssh できることを確認する

初回実行時は[y]入力が必要になる

```
$ ssh master0 (Hadoop のとき master1)
```

```
$ ssh slaveXXX
```

【補足 1】以下の場合に公開鍵の再配布が必要となる

```
$ sshpass -p "tpu" ssh-copy-id -i ~/.ssh/id_rsa.pub -o "StrictHostKeyChecking no" master0
```

1) 別のマスターPC を slave で使用する場合

2) slave 間で master の ssh-key が異なっており、円周率の並列処理が上手くできない場合

【補足 2】それぞれの対処法

1)

- ・各 slave で 53 と 62 の再設定を学生に促す
- ・ssh-key が作られている場合同様に key を配る

```
sshpass -p "tpu" ssh-copy-id -i ~/.ssh/id_rsa.pub -o "StrictHostKeyChecking no" slaveXXX
```

- ・作られていない場合(おそらく↑でエラーが出る)新規作成して再配布

```
ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
```

2)

- ・マスターPC の「~/.ssh/known_hosts」でグループの slave の情報を削除する
- ・slave(グループ全員分)の known_hosts でも master 番号の key 情報を削除する
- ・key の再配布を行う(sshpass ~ slaveXXX)

T-3. 終わり

T-4.MasterPC の MPICH の初期設定(MPICH をダウンロードした後一度だけ必要)

共有ディレクトリの作成（作ってあつたら必要なし）

```
$ mkdir /home/tpu/share
```

共有ディレクトリの設定

```
$ sudo vim /etc/exports
```

↓を最後の行に追加

```
/home/tpu/share 192.168.11.1/24(rw,no_root_squash)
```

ドメイン名の設定

```
$ sudo vim /etc/idmapd.conf
```

↓の Domain のコメントアウトをなくす

[General]

```
#Verbosity = 0
```

```
# The following should be set to the local NFSv4 domain name
```

```
# The default is the host's DNS domain name.
```

```
Domain = local.domain.edu
```

T-4.MasterPC の MPICH の初期設定(MPICH をダウンロードした後一度だけ必要)(終わり)

58. マスターPCに通信できるか確かめる。

```
$ ping masterX (X: master 番号)
```

以下のように表示されていれば通信できている。

```
64 bytes from master1 (192.168.11.20X): icmp_seq=1 ttl=64 time=0.111 ms
```

通信できていない場合以下を確認してみる

- LANケーブルが接続されているか確認する。
- 仮想マシンの右上の電池のマークのところをクリックして Ethernet(enp0s3)が接続状態になっていることを確認する。
- 仮想マシンの右上の電池のマークのところをクリックして右下の電源ボタンから電源オフを選択し、もう一度起動する。
- マスターPCの方でも上記3つを実行する。
- これでできなかった場合はTAを呼ぶ。(ルーターの差し直し?)

59. スレーブPCがマスターPCをマウントできるように以下のコマンドをマスターPCで入力する。

※マウントとは、スレーブPC(生徒側のPC)がマスターPCの特定のフォルダを操作・利用可能にすること。

```
$ sudo systemctl start rpcbind
$ sudo systemctl start nfs-server
$ sudo systemctl start nfs-lock
$ sudo systemctl start nfs-idmap
$ sudo systemctl enable nfs-server
$ sudo systemctl reload nfs-server
```

60. 自分のPCに戻り(「exit」でsshを解除する)、マスター側の共有ディレクトリをマウントするディレクトリを、以下のコマンドで作成する。

```
$ mkdir /home/tpu/share
```

61. slave側で、以下のコマンドを打ってマウントする。(X: 使っている master 番号)

slave側で操作する必要があるため、[tpu@slaveXX]になっているか確認する

```
$ sudo mount --types nfs masterX:/home/tpu/share /home/tpu/share
```

T-x. access denied の場合

「mount.nfs: access denied by server while mounting masterX:/home/tpu/share」 と出る場合、以前の実験の mount 情報が残っているため、master 側の share フォルダの名前を変更し(2 など)、新しい share フォルダを作成する(mkdir /home/tpu/share)

T-x.(終わり)

以下のコマンドで確認し、

```
$ df -k
```

一番下に次のように表示されればいい

```
192.168.11.20X:/home/tpu/share 430897664 847872 430049792 1% /home/tpu/share
```

62. slave 側で以下のコマンドで共有ディレクトリを作成するために fstab ファイルを編集する。

```
$ sudo vim /etc/fstab
```

最後の行に以下を追加する。「share」の後は改行せずスペースだけ。

```
192.168.11.20X:/home/tpu/share /home/tpu/share
```

```
nfs rsize=8192,wszie=8192,hard,intr 0 0
```

63. 確認方法

```
$ sudo umount /home/tpu/share
```

```
$ sudo mount -a
```

```
$ df -k
```

一番下につぎのようにでればいい。

```
192.168.11.20X:/home/tpu/share 430897664 847872 430049792 1% /home/tpu/share
```

T-y. マウント時のエラー対処

(1) 「mount -a」の応答が帰ってこない slave がある場合、59 の nfs を再実行

→59 の一部ができなくなっているときは間違えて master の fstab を 62 で書き換えてしまった可能性があるので、元に修正

(2) 後の share フォルダで別 master のマウントが判明した場合

- 63 「umount」でマウント解除

- それでも「master is busy」と出る場合は、slave の仮想マシンを再起動する

→デスクトップの share がなくなり解除される

T-y.(終わり)

64. マスターPC に SSH で接続する。

```
$ ssh masterX
```

65. host ファイルを書き換える(マスターPC に SSH 接続したまま)。

この host ファイルの中に記述されている PC を用いて並列処理が行われる

→X は自分が接続する master 番号

```
$ sudo vim /home/tpu/share/host
masterX
slave5
```

(エラーが出たとき)

Cd /home/tpu/share/host を入力

Ls -a を入力

sudo rm .host. swp を入力

(終わり)

(並列する pc を変更したいとき)

Slave の番号を変える。または増やす

※同時に並列処理する台数に応じて、この部分を変更する(課題 2~4 の場合)

(上の例はマスターPC と自分の PC(学籍番号 5)との並列処理の場合)

66. 自分の学籍番号の下二桁が 35 の場合

以下の内容のプログラムを montecarlo35.c という名前で作成する。コピペ可。

コピペ方法：共有ディレクトリの share の中に montecarlo.c があるのでそれをコピーして貼り付けし(同じフォルダ内で OK)、名前を montecarlo35.c にする。(推奨)

数値積分やモンテカルロ法を用いて円周率を求めるサンプルプログラム

(Mpich)

```
#include "mpi.h"
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

#define NUM_MAX 100000000
```

```

#define PI 3.141592653589793238462643

typedef struct
{
    double x;
    double y;
}POINT;

int main(int argc, char* argv[])
{
    POINT p;
    double r, pi, s, e;
    int cnt, total_cnt, i, rank, size, local_cnt;
    int start, end, sendbuf, recvbuf;

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    s=MPI_Wtime();

    local_cnt=(int)((double)NUM_MAX/(double)size);

    start=rank*local_cnt;
    end=(rank+1)*local_cnt;

    srand((unsigned int)time(NULL)*(rank+1));
    cnt = 0;

    for (i = start; i < end; i++){
        p.x = (double)rand() / (double)RAND_MAX;
        p.y = (double)rand() / (double)RAND_MAX;

        r = sqrt(p.x * p.x + p.y * p.y);

        if ( r < 1.0 ){
            cnt++;
        }
    }
}

```

```

    }

}

sendbuf = cnt;
recvbuf = 0;

MPI_Allreduce(&sendbuf, &recvbuf, 1, MPI_INT, MPI_SUM, MPI_COMM_WORLD);
total_cnt =recvbuf;

pi = 4.0 * (double)total_cnt / (double)NUM_MAX;

if(rank==0){

    printf("pi: %18.16lf, Error: %18.16lf\n",pi, fabs(pi-PI));
    e=MPI_Wtime();
    printf("time = : %8.4f[sec]\n",e-s);
}

MPI_Finalize();

return 0;
}

```

67. 以下のコマンドで共有フォルダに移動する。

```
$ cd /home/tpu/share
```

68. 端末上で以下のコマンドでコンパイルする。

```
$ mpicc montecarlo35.c -o montecarlo35 -lm
```

69. 端末上で以下のコマンドで実行する。班の人と mpirun コマンドを同時に実行しないようにする。
-*np* の後の数字で、使う PC の数が決まる。つまり、-*np* 2 だと 2 台、-*np* 1 だと 1 台で実行される。

```
$ mpirun --hostfile host -np 1 ./montecarlo35
```

結果の見方

pi がモンテカルロ法によって導き出された円周率の値で、Error が正確な円周率の値との差が表示されている。そして、time が求めるまでかかった時間を表す。

```
[tpu@master2 share]$ mpicc montecarlo00.c -o montecarlo00 -lm
[tpu@master2 share]$ mpirun --hostfile host -np 1 ./montecarlo00
rank=0,start=0,end=100000000
rank=0,cnt=78540047
78540047rank=0,start=0,end=100000000
pi: 3.1416018800000001,Error: 0.0000092264102070
time = 5.345662[s]
```

T-5. 手順 69 番目のエラー対処

2台以上実行時「private key files are NOT accessible by others」のエラーが発生するとき

```
[tpu@master3 share]$ mpirun --hostfile host -np 2 ./montecarlo70
@           WARNING: UNPROTECTED PRIVATE KEY FILE!           @
@           Permissions 0664 for '/home/tpu/.ssh/id_rsa' are too open.
@           It is required that your private key files are NOT accessible by others.
@           This private key will be ignored.
Load key "/home/tpu/.ssh/id_rsa": bad permissions
Permission denied, please try again.
Permission denied, please try again.
Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

Master3 側の鍵が、他 slave PC にアクセスされすぎると生じる。

【対処】

1) /home/tpu/.ssh ディレクトリにある「id_rsa」、「id_rsa.pub」、「known_hosts」を削除する。

```
$ cd /home/tpu/.ssh
```

```
$ rm id_rsa
```

2) ssh-key の作成

```
$ ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
```

ssh で入るマスター本体を操作して、並列計算する slave に鍵を配り直す

・1回のみ

```
$ sshpass -p "tpu" ssh-copy-id -i ~/.ssh/id_rsa.pub -o "StrictHostKeyChecking no" master
```

・ slave ごとに

```
$ sshpass -p "tpu" ssh-copy-id -i ~/.ssh/id_rsa.pub -o "StrictHostKeyChecking no" slaveXXX
```

T-5.(終わり)

70. 計算する台数を増やしながらレポート課題を進めていく。

71. **1週目のレポート課題を終える。** 以上で、1週目における作業は終了です。

72. 最終的に全ての作業が問題なくできたかどうかは、TA が master から全員の slave 上で問題なく分散ネットワーク処理が実行できか確認しますので、うまくいくまで待っていてください。また、やり忘れた作業など、追加の作業の指示が TA からあった場合には、都度対応してください。
73. 組の全員の仮想マシンが無事 slave として動作していることが確認できたら、1 週目の実験自体は終了です。ただし、2 週目の最初に全く同じ状況を元通りに構築する必要があるため、最後に仮想マシンを正常にシャットダウンさせる必要がありますので、下記の作業を行ってください。
74. 仮想マシンをシャットダウンします。以下のコマンドを実行します。
\$ poweroff
75. VirtualBox の仮想ウィンドウが閉じ、仮想マシンが停止状態になったのを確認したら、VirtualBox を終了します。
76. Windows を終了します。

参考文献

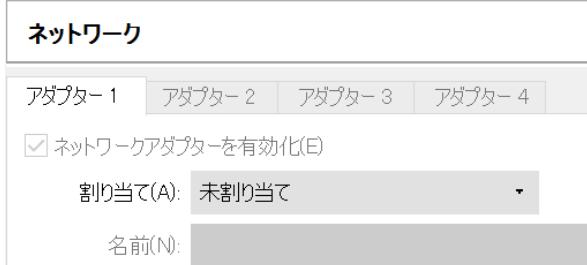
※以下に挙げているものは、主に1週目における Hadoop の構築についての参考書であり、分散ネットワークの参考文献ではない点に注意すること

- [1] 「Hadoop 第3版」 オライリージャパン
- [2] 「Hadoop徹底入門 第2版 オープンソース分散処理環境の構築」 翔泳社

S1. VirtualBox が起動できない(ネットワーク未割り当て)

○起動時、勝手にネットワーク割り当てが「未割り当て」にされ、起動できなくなることがある

→「ブリッジアダプター」を選択しても「名前」の部分に「Wi-Fi 名」が出てこない



※PC でいつも使われているアダプター(おそらく「Intel(R) Wireless-AC 9560 160MHz」)

に VirtualBox のドライバが入っていない(認識していない)のが原因とのこと

<https://lineagewalker.blogspot.com/2019/06/VirtualBox-BridgeAdapter.html>

(1)コントロールパネル「ネットワークと共有センター」→「アダプター設定の変更」を選択すると
「ネットワーク接続」が出てくるので、「Wi-Fi」を右クリック→「プロパティ」へ移動



(2-1)「VirtualBox」にチェックがなければ入れて「OK」で

VirtualBox を再起動してみる

項目自体がなければ、「インストール」で追加

(2-2)別ウィンドウが出てくるので「サービス」→「追加」

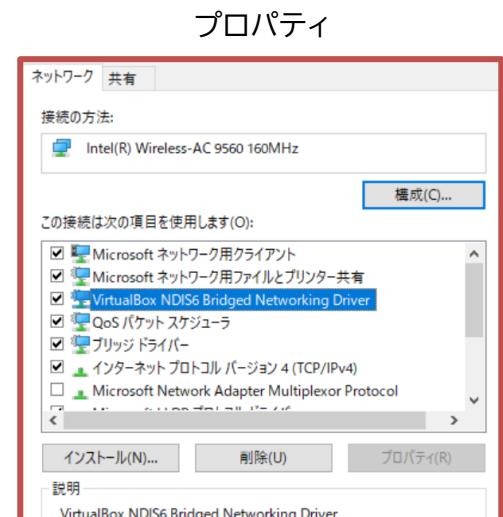
→「ディスク使用」で「フロッピーディスクからインストール」を開く

(3) 参照で VirtualBox ドライバのディレクトリを辿っていく

おそらく、

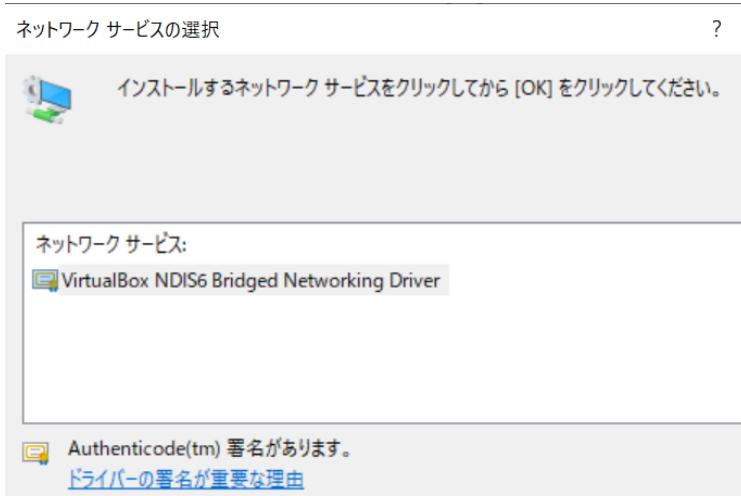
「C:\Program Files\Oracle\VirtualBox\drivers\network\netlwf」に行くと

「VBoxNetLwf.inf」があるのでダブルクリックして「OK」を選択



名前	更新日時	種類
VBoxNetLwf.inf	2020/10/16 17:31	セットアップ情報

(4)下図のサービスを選択して「OK」にする



(5)「VirtualBox NDIS6 ~」にチェックが入っていれば、再起動で VirtualBox は通常起動される」

S2. VirtualBox が起動できない(起動したが、ログイン画面までいかない)

○以下の文字が画面に表示される場合

<https://thr3a.hatenablog.com/entry/20180712/1531402449>

*** An error occurred during the file system check.

*** Dropping you to a shell; the system will reboot

*** when you leave the shell.

Give root password for maintenance

(or type Control-D to continue):

※手順 62.「fstab の編集」でマウント関連のエラーがあるために起こる

(1)ログイン(root 権限)パスワードが求められているので、入力→コマンド画面へ

(2)「mount -o remount,rw /」を入力

(3)「vim /etc/fstab」でマスター番号が異なっていると思われる所以、修正

(4)「reboot」によって正常起動する

S3. HADOOP の並列処理実行時のエラー

→「YarnException: Unauthorized request to start container」

○以下が表示され、下線部の文言が出てくる場合

<https://stackoverflow.com/questions/20257878/yarnexception-unauthorized-request-to-start-container>

map 100% reduce 0%

13/11/28 09:57:15 INFO mapreduce.Job: Task Id : attempt_1385611768688_0001_r_000000_0, Status : FAILED

Container launch failed for container_1385611768688_0001_01_000003 : org.apache.hadoop.yarn.exceptions.

YarnException: Unauthorized request to start container.

This token is expired. current time is 1385612996018 found 1385612533275

at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)

→1385612996018 秒=2013 年 11 月 28 日 13 時 29 分 56.018 秒 の現在時刻に対し、

仮想マシンが 1385612533275 秒=〃 13 時 22 分 13.275 秒となっており、ズレていることが記載

時間参考「<https://keisan.casio.jp/exec/system/1526004418>」

※各スレーブの現在時刻が違うために、起こっているとのこと

(1)スレーブ間を同期させる際のタイムアウト時間(デフォルト 10 分)を延ばすことでエラーを回避する

(2)slave 側で、「`sudo vim hadoop-2.8.5/etc/hadoop/yarn-site.xml`」を実行し、時間延長のプロパティを新たに追加する(今回は 600 秒→1000 秒に延長)

```
</property>
<name>yarn.resourcemanager.rm.container-allocation.expiry-interval-ms</name>
<value>1000000</value>
</property>
```

以下のように末尾に追加する

```
<property>
  <name>dfs.namenode.datanode.registration.ip-hostname-check</name>
  <value>false</value>
</property>
<property>
  <name>yarn.resourcemanager.rm.container-allocation.expiry-interval-ms</name>
  <value>1000000</value>
</property>
</configuration>
```