

0910追記

- I have no name!となる問題を解決しました.
- ユーザでコンテナを起動した際にpythonなどのコマンドが使えない問題を解決しました.

Dockerの利用に関して(問題)

- Dockerでコンテナ(仮想環境)を作成した場合, デフォルトではrootの状態起動される
- コマンドの例

コンテナ側

```
root@fe24b5ded00a:/workspace# sleep 1000
```

ホスト側

```
root      338368  0.0  0.0   4384   648 pts/0    S+   14:01   0:00 sleep 1000
kojima    338369  0.0  0.0   21164  3288 pts/0    R+   14:01   0:00 ps aux
```

誰が実験をしているか把握できない

Dockerの利用に関して(問題)

- Dockerでコンテナ(仮想環境)を作成した場合, デフォルトではrootの状態起動される
- ファイル管理の例

コンテナ側

```
root@fe24b5ded00a:/workspace# echo root > root.txt
```

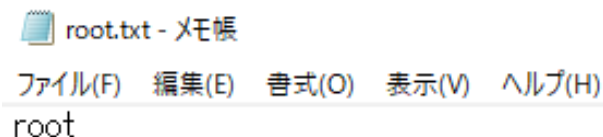
ホスト側

```
-rw-r--r-- 1 root root 5 9月 2 14:05 root.txt  
-rw-r--r-- 1 kojima users 2594 8月 19 16:32 test_kitti_depth.py
```

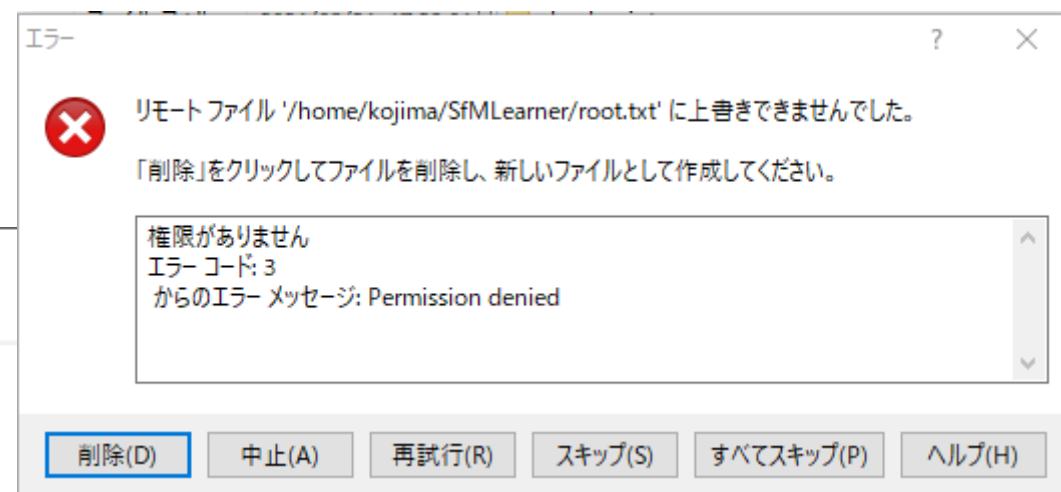
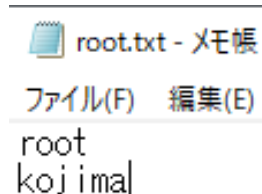
Dockerの利用に関して(問題)

- Dockerでコンテナ(仮想環境)を作成した場合, デフォルトではrootの状態で起動される

- ファイル管理の例



編集&保存



ファイルを作る場所によっては削除もできない

Dockerの利用に関して(問題)

- Dockerでコンテナ(仮想環境)を作成した場合, デフォルトではrootの状態では起動される
- その他の問題 . . . root権限でsudo権が使える

```
root@fe24b5ded00a:/workspace# sudo ls
LICENSE      SfMLearner.py  __pycache__    data           demo.ipynb     kitti_eval     models         test_kitti_depth.py  train.py
README.md    SfMLearner.pyc checkpoints     data_loader.py formatted       misc           nets.py        test_kitti_pose.py   utils.py
```

もちろんホスト側では使えない

```
kojima@sophie:~/SfMLearner$ sudo ls
[sudo] password for kojima:
kojima is not in the sudoers file.  This incident will be reported.
```

Dockerの利用に関して(解決法)

- docker runもしくはdocker execコマンドの-uオプションでユーザを指定してコンテナを起動することができる

```
kojima@sophie:~/SfMLearner$ whoami
kojima
kojima@sophie:~/SfMLearner$ id kojima
uid=1006(kojima) gid=100(users) groups=100(users),998(docker)
```

↑ この場合uid=1006,gid=100

```
docker run -it -u=1006:100 --gpus all --name usertest -v /home/kojima/SfMLearner:/workspace cuda8_ubuntu16_img
```

or

```
docker exec -it -u=1011:100 3652f6dade97 /bin/bash
```

↑ docker起動時に-u={uid}:{gid}で指定して起動

I have no name!となった場合①

- I have no name!とならずにdockerをユーザ指定して起動する方法

```
sudo docker run -it --gpus all --name cu10kojima_kojima -v /home/kojima/sfm_flow2021:/workspace -v /etc/group:/etc/group:ro -v /etc/passwd:/etc/passwd:ro -u=1011:100 cudakojima
```

上記コマンドのように-vコマンドを用いて

- /etc/group
- /etc/passwd

をマウントする。こうすることでホスト側とコンテナ側でユーザ情報が共有される。

※:roは読み取り専用という意味

I have no name!となった場合②

- I have no name!とならずにdockerをユーザ指定して起動する方法(推奨)

```
sudo docker run -it --gpus all --name cu10kojima_kojima -v /home/kojima/sfm_flow2021:/workspace -v /etc/group:/etc/group:ro -v /etc/passwd:/etc/passwd:ro cudakojima
```

上のコマンドのようにユーザを指定せずにrootで起動してから、ユーザに切り替えることも可能

その場合”sudo docker exec -it -u=1011:100 3652f6dade97 /bin/bash”のようにexecコマンドの-uオプションでユーザを指定することができる.

推奨するのは一旦rootユーザとしてdockerを起動し、環境を作ってから、ユーザを指定してコンテナに入り直す方法です(②の方法)

一般ユーザで入ってしまうと環境を作るときに権限関係でエラーが出る可能性があるので...

pythonなどのコマンドが使えない時

まずPATHなどが通っているかの確認

→のdockerfileを仮定する

- 赤枠について

“bash Anaconda3-2019.10-Linux-x86_64.sh -b” はデフォルトではroot以下にファイルが作成されるのでユーザでコンテナに入ると権限がなく、python系のコマンドが使えない

-p “インストール先” で権限がなくてもアクセスできるインストール先を指定することでpython系のコマンドが使用可能になる

- 黄枠について

赤枠で指定した先にPATHを通す

これで大体は大丈夫なはず

```
FROM nvidia/cuda:10.0-cudnn7-devel-ubuntu16.04

# update packages
RUN set -x && \
    apt-get update && \
    apt-get upgrade -y

# install command
RUN set -x && \
    apt-get install -y wget && \
    apt-get install -y sudo

# anaconda
RUN set -x
RUN wget https://repo.anaconda.com/archive/Anaconda3-2019.10-Linux-x86_64.sh
RUN bash Anaconda3-2019.10-Linux-x86_64.sh -b -p /kk
RUN rm Anaconda3-2019.10-Linux-x86_64.sh

# path setting
ENV PATH $PATH:/kk/bin

RUN conda install pytorch==1.0.1 torchvision==0.2.2 cudatoolkit=10.0 -c pytorch

#RUN apt install -y python3-pip
#RUN python3 -m pip install --upgrade pip

WORKDIR /home/kojima/sfm_flow2021
#RUN conda update --all
ADD requirements.txt /home/kojima/sfm_flow2021
RUN pip install -r requirements.txt
#RUN conda install -c conda-forge argparse
#RUN conda install -c conda-forge tensorboardx
#RUN conda install -c conda-forge blessings
#RUN conda install -c conda-forge progressbar2
#RUN conda install -c conda-forge tqdm
#RUN conda install -c conda-forge pebble

# move to root directory
WORKDIR /home/kojima
#RUN pip3 install torch torchvision
```

pythonなどのコマンドが使えない時

2021/09/10時点

- python系列などpipでインストールするものは一般ユーザではインストールエラーになるが、rootの状態ですべてインストールして一般ユーザに切り替えれば問題なく使えた
- C++などのモジュールをインストールするときにインストール先がrootなどのアクセス権がない場所にインストールされる場合、一般ユーザはそのモジュールを使えないことがある

参考サイト

- <https://unskilled.site/docker%E3%82%B3%E3%83%B3%E3%83%86%E3%83%8A%E5%86%85%E3%81%AE%E3%83%A6%E3%83%BC%E3%82%B6%E3%81%A8%E3%83%9B%E3%82%B9%E3%83%88%E3%81%AE%E3%83%A6%E3%83%BC%E3%82%B6%E3%81%A8%E3%81%AE%E9%96%A2%E4%BF%82/>
- <https://qiita.com/manabuishiirb/items/83d675afbf6b4eea90e4>
- <https://docs.docker.com/engine/reference/run/>