System Specifications

- **Machine Model:** [NVIDIA reComputerJ10]
- **Label:** [Machine #1]

Overview

This document serves as comprehensive documentation for all the tasks and configurations done on my NVIDIA machine.

Table of Contents

0. System Requirements and Setup
1. USB Port C Observations
2. Attempts to Install Conda Through Miniforge and Mambaforge on an SD Card
3. How to Make New Conda Environments be Stored on the SD card by Default
4. Properly uninstalling Miniforge-pypy3 and Installing Miniforge 3
5. Installing Segment Anything

0. System Requirements and Setup

0.1 System Configuration

Description

The following text documents the actions I performed while navigating the system configuration.

L4T_End_User_License_Agreement:

Select "I accept the terms of these licenses"

Language:

Select "English""

Where are you?:

Select "Los Angeles""

Who are you?:

Your Name: UCSBOEP Your computer's name: ucsboep-desktop Pick a username: ucsboep Password: diffusion.1846

Select Nvpmodel Mode:

Select "Maxn-Default""

Would you like to set up Livepatch now?:

Select "Set up live patch""

Help improve ubuntu: Would you like to send this information:

Select "Yes, send system info to Canonical""

0.2. Required Materials

To ensure your NVIDIA machine functions properly, you'll need the following materials and accessories:

1. **WiFi Adapter** Attach the WiFi Adapter to your NVIDIA Machine. It is called "TP-Link AC600 USB WiFi Adapter for PC". After you've attached it you should see a WiFi icon in the upper right corner. Please be sure to look for the WiFi you'd like to connect to and enter its password.

2. **Monitor** Use the HM Tech Display for Raspberry Pi. It should come with a stand and power cord.

3. **Power Cord for the Monitor with USB-A Port** The monitor requires power to function. You can connect the monitor to an outlet using the provided power cord with a USB-A port.

4. **HDMI to HDMI Cord** To establish a video connection between your machine and the monitor, use an HDMI to HDMI cord. Connect one end to your device and the other end to the monitor's HDMI port.

5. **Logitech Wireless Keyboard** Use a Logitech wireless keyboard. Connect the wireless USB A port to your machine to pair it.

6. **Power Cord with USB-C Port** Connect your machine to a power source using the provided power cord with a USB-C port. Ensure the power cord is plugged into the correct port, there are two USB-C ports, one for data and one for power.

7. **SD Card and SD Card Reader with USB-A Port** An SD card and an SD card reader are useful for data storage and transfer. Insert the SD card into the reader and connect the reader to a USB-A port on your machine.

These materials are essential for setting up and using your NVIDIA machine!

1. USB Port C Observations

1.1 USB-C Port Doesn't Read SD Cards

Description:

I've observed that the USB-C port on my machine is not functioning as expected when it comes to reading SD cards. When I insert an SD card into the USB-C port; I notice it's not recognized by the system. This issue has been consistent across multiple SD cards I've tested.

Potential Causes:

1. **Driver Issue:** There might be a missing or outdated driver for the USB-C port, causing it to not properly communicate with SD cards.

2. **Physical Connection:** There could be a loose or damaged connection within the USB-C port that is affecting its ability to read SD cards.

Troubleshooting Steps Taken:

1. I've tested the same SD cards in other ports (e.g., USB-A) on my machine, and they are being recognized without any issues.

2. I've mounted and unmounted it several times through the command line and physically

3. I've used commands 'lsblk' and 'fdisk -l' to see if it is detected. It was not detected.

Next Steps:

1. I'll consider reaching out to technical support for further assistance.

2. Attempts to Install Conda Through Miniforge and Mambaforge on an SD Card

Description: When I received the NVIDIA machine there was only around 700 MB of storage left. This meant I was unable to create a conda environment for the segment anything model, a model I was planning to use. First, I tried using an SD card for external storage to solve this issue. I attempted to install conda on the SD card, but unfortunately, I encountered denied permissions and several other limitations with the SD card. I've documented my steps below.

2.1 Attempt 1

Description: I tried to install multiple versions of mambaforge and miniforge from this website These attempts were unsuccessful.

0. Plug in the SD card

1. Download Mambaforge for ARM: here

Download installer "Mambaforge-23.3.1-0-Linux-aarch64.sh".

2. Transfer Mambaforge Installer to SD Card:

Transfer the downloaded Mambaforge installer to the SD card using the `mv` command. The SD card is mounted at the path "/media/ucsboep/NVIDIASD". Make sure you're in the directory of the downloaded installer. If you downloaded it from the website it should have been put in /home/ucsboep/Downloads.

```
cd /home/ucsboep/Downloads
mv Mambaforge-23.3.1-0-Linux-aarch64.sh /media/ucsboep/NVIDIASD
```

3. Create Mambaforge Folder:

Create a folder named "mambaforge" on the SD card. It is good practice to have a specific folder for conda.

```
mkdir /media/ucsboep/NVIDIASD/mambaforge
```

4. Move Installer to Mambaforge Folder:

Move the Mambaforge installer into the "mambaforge" folder:

```
mv /media/ucsboep/NVIDIASD/Mambaforge-23.3.1-0-Linux-aarch64.sh
/media/ucsboep/NVIDIASD/mambaforge
```

5. Navigate to Mambaforge Folder:

Navigate to the "mambaforge" folder on the SD card:

```
cd /media/ucsboep/NVIDIASD/mambaforge
```

6. Make Mambaforge Installer Executable:

```
chmod +x Mambaforge-23.3.1-0-Linux-aarch64.sh
```

7. Run Mambaforge Installer:

```
./Mambaforge-23.3.1-0-Linux-aarch64.sh
```

ERROR

Permission Denied

2.2 Attempt 2

Description: My steps in this attempt are identical to attempt 1 up until step 8. For this attempt, I used the sudo command to get permission.

Remove all previous installers and folders created using:

```
rm -r /path/to/folder/with/mamba
```

(i.e. rm -r /media/ucsboep/NVIDIASD/mambaforge)

0. Plug in SD card

1. Download Mambaforge for ARM: here

Download installer "Mambaforge-23.3.1-0-Linux-aarch64.sh".

2. Transfer Mambaforge Installer to SD Card:

Transfer the downloaded Mambaforge installer to the SD card using the `mv` command. The SD card is mounted at the path "/media/ucsboep/NVIDIASD". Make sure you're in the directory of the downloaded installer. If you downloaded it from the website it should have been put in /home/ucsboep/Downloads.

```
cd /home/ucsboep/Downloads
mv Mambaforge-23.3.1-0-Linux-aarch64.sh /media/ucsboep/NVIDIASD
```

3. Create Mambaforge Folder:

Create a folder named "mambaforge" on the SD card. It is good practice to have a specific folder for conda:

```
mkdir /media/ucsboep/NVIDIASD/mambaforge
```

4. Move Installer to Mambaforge Folder:

Move the Mambaforge installer into the "mambaforge" folder:

```
mv /media/ucsboep/NVIDIASD/Mambaforge-23.3.1-0-Linux-aarch64.sh
/media/ucsboep/NVIDIASD/mambaforge
```

5. Navigate to Mambaforge Folder:

Navigate to the "mambaforge" folder on the SD card:

```
cd /media/ucsboep/NVIDIASD/mambaforge
```

6. Make Mambaforge Installer Executable:

```
chmod +x Mambaforge-23.3.1-0-Linux-aarch64.sh
```

7. Run Mambaforge Installer:

```
sudo  ./Mambaforge-23.3.1-0-Linux-aarch64.sh
```

8. Follow prompts and press enter to continue:

9. Accept license terms:

Here is a snippet of the terminal at this step:

Do you accept the license terms? [yes|no] [no] >>> yes

Mambaforge will now be installed in this location: /home/ucsboep/mambaforge

- Press ENTER to confirm the location

- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/ucsboep/mambaforge] >>>

I chose to specify a different location. Here is what I typed:

[/home/ucsboep/mambaforge] >>> `/media/ucsboep/NVIDIASD/mambaforge`

ERROR

File location already exists

[/home/ucsboep/mambaforge] >>> /media/ucsboep/NVIDIASD/mambaforge ERROR: File or directory already exists: '/media/ucsboep/NVIDIASD/mambaforge' If you want to update an existing installation, use the -u option. ucsboep@ucsboep-desktop:/media/ucsboep/NVIDIASD/mambaforge$

I decided to create a new file in this step instead of making the directory at the beginning to try to solve this issue.

1. Make Mambaforge Installer Executable:

```
chmod +x Mambaforge-23.3.1-0-Linux-aarch64.sh
```

2. Run Mambaforge Installer:

```
sudo  ./Mambaforge-23.3.1-0-Linux-aarch64.sh
```

3. Follow the prompts and press enter to continue:

4. Accept license terms:

Here is a snippet of the terminal at this step:

Do you accept the license terms? [yes|no] [no] >>> yes

"Mambaforge will now be installed into this location: /home/ucsboep/mambaforge

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/ucsboep/mambaforge] >>> /media/ucsboep/NVIDIASD/mambaforge_new PREFIX=/media/ucsboep/NVIDIASD/mambaforge_new Unpacking payload ... Extracting ca-certificates-2023.7.22-hcefe29a_0.conda critical libmamba Can't create 'ssl/cert.pem' ucsboep@ucsboep-desktop:/media/ucsboep/NVIDIASD/mambaforge$

According to my research, the reason this didn't work was because I had to use sudo to bypass permission denied. The use of sudo automatically only gave me permission in the root folder. This explains why it kept trying *to place mambaforge in /home/ucsboep/mambaforge. I believe this is why it didn't allow me to install conda onto the SD card.*

I thought this may be because I don't have read-and-write permissions on the SD card.

2.3 Attempt 3

Description: This time I focused on seeing if I could bypass permission denied without the use of sudo. It seemed that sudo was the reason I was unable to install conda onto the SD card.

1. First I checked the filesystem for read and write permissions on the SD card:

```
mount
ls -ld /media/ucsboep/NVIDIASD
```

This snippet from the terminal showed I have read and write permissions on the sd card, therefore this was not the issue:

/dev/sda1 on /media/ucsboep/NVIDIASD type vfat (rw,nosuid,nodev,relatime,uid=1000,gid=1000,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,showexec,utf8,flush,errors=remount-ro,uhelper=udisks2)

2. Next, I mounted and unmounted it to see if permissions change after this:

```
sudo umount /media/ucsboep/NVIDIASD
sudo mount /dev/sda1 /media/ucsboep/NVIDIASD
```

3. I ran all the steps in attempt 1 and got permission denied again.

4. Other attempts:

I used the - u options suggested in the error for attempt 2. Unfortunately, I still received an error.

Here is a snippet from my terminal

ERROR

File location already exists

[/home/ucsboep/mambaforge] >>> /media/ucsboep/NVIDIASD/mambaforge ERROR: File or directory already exists: '/media/ucsboep/NVIDIASD/mambaforge' If you want to update an existing installation, use the -u option. ucsboep@ucsboep-desktop:/media/ucsboep/NVIDIASD/mambaforge$

I also tried installing miniforge3 and mambaforge but ran into the same issue

3. How to Make New Conda Environments be Stored on the SD card by Default

Description: After learning I didn't have the permissions to run conda on an SD card, I tried to keep the conda installed at the root and change it so that new conda environments were stored on the SD card instead of the internal storage. This is another attempt to increase storage on the NVIDIA machine. I had miniforge-pypy3 installed at the root when I made these attempts. The conda version I had installed at the root when I made these attempts. In section 4 I will describe how and why I deleted miniforge-pypy3 and installed miniforge3 instead.

All steps were reversed after this attempt so new conda environments aren't stored in the SD card by default anymore.

3.1 Temporarily changing the location

Description: The following steps describe how to temporarily make the SD card hold all new conda environments folders. Since it is temporary, this means it only applies to the current session. I demonstrate how to permanently change it in section 3.2 so that all future versions of conda installed on the NVIDIA machine have their environments stored on the SD card. These attempts were semi-successful. I was able to successfully create some environments as long as I did not specify a python version different from the version already in the environment. For example: 'conda create --name myenv' works and when you check the version of python it says python version 2.7. But if you run 'conda create --name myenv python=3.8' you run into an error. Additionally, I'm unable to install any other versions of python in this environment. Perhaps, this will work well for environments that solely require python version 2.7, but I run into issues when I try using other versions.

0. Plug in SD card

1. Check Mount Point of SD Card Verify that your SD card is properly mounted and accessible. You can use the `df -h` command to list the available filesystems and their mount points.

2. Set `CONDA_ENVS_PATH` Environment Variable Make sure you're in the home directory first by using cd ~. Then set the `CONDA_ENVS_PATH` environment variable to the path on your SD card where you want to create new Conda environments. Replace `/media/ucsboep/NVIDIASD` with the actual path to your SD card. Here NVIDIASD is the name of the SD card and envs is the name of a folder in the NVIDIASD SD card where I am storing all new environments. This command creates the folder 'envs', but feel free to change it to any other folder name.

```
cd ~
export CONDA_ENVS_PATH=/media/ucsboep/NVIDIASD/envs
```

3. Activate Miniforge Environment Activate the Miniforge environment by navigating to the directory where Miniforge is installed and sourcing the activation script.

```
cd /home/ucsboep/miniforge-pypy3
source bin/activate
```

4. Create New Conda Environment on SD Card Use the `conda create` command to create a new Conda environment on your SD card. Replace `<environment_name>` with your desired environment name.

```
conda create --name myenv
```

5. Activate and Use the New Environment Activate the newly created environment to work within it.

```
conda activate myenv
```

6. Deactivate Environment Deactivate the environment when you're done working in it.

```
conda deactivate
```

By following these steps, you've successfully configured Conda to create and manage environments on your SD card temporarily, which helps save space on your computer's internal storage.

3.2 Permanently changing the location

Description: The following steps describe how to permanently make the SD card hold all new conda environment folders. Since it is permanent, this means all future versions of conda installed on the NVIDIA machine have their environments stored in the SD card. These attempts were semi-successful. I was able to successfully create some environments as long as I did not specify a python version different from the version already in the environment. For example: 'conda create --name myenv' works and when you check the version of python it says python version 2.7. But if you run 'conda create --name myenv python=3.8' you run into an error. Additionally, I'm unable to install any other versions of python in this environment. Perhaps, this will work well for environments that solely require python version 2.7, but I run into issues when I try using other versions.

0. Plug in the SD card

1. Check Mount Point of the SD Card Verify that your SD card is properly mounted and accessible. You can use the `df -h` command to list the available filesystems and their mount points.

2. Navigate to Home Directory

```
cd ~
```

3. Create .bashrc File

```
touch ~/.bashrc
```

4. Edit .bashrc File

```
gedit ~/.bashrc
```

5. Add Environment Variable Inside the `.bashrc` file, add the following line to set the environment variable `CONDA_ENVS_PATH` to your desired directory:

```
export CONDA_ENVS_PATH=/media/ucsboep/NVIDIASD/envs
```

6. Save the changes and close the `.bashrc` file: Press 'Ctrl' + 'S' to save the file Press 'Ctrl' + 'Q' to close the editor
7. Reload .bashrc To apply the changes immediately in your current terminal session, run:

```
source ~/.bashrc
```

8. Verification To verify that the change took effect, check the value of the `CONDA_ENVS_PATH` variable:

```
echo $CONDA_ENVS_PATH
```

By following these steps, you successfully changed the default location for Conda environments and made the change permanent. Any new Conda environments you create will now be stored in the specified directory on your SD card.

3.3 Creating a conda environment in the envs folder on the SD card without specifying the python version

Description: The following steps describe how to create new conda environments in the SD card. However, as mentioned in sections 3.1 and 3.2. This only works when you create an environment without specifying a python version different from python 2.7. By default, the version installed with this conda version is python 2.7.

1. Make sure you are in the correct directory where Miniforge is installed.

```
cd /home/ucsboep/miniforge-pypy3
```

2. Activate Miniforge: Activating the base environment:

```
source bin/activate
```

If this command succeeds, you should see `(base)` at the beginning of your terminal prompt.

3. Create a new conda environment

```
conda create --name myenv
```

4. To activate this environment use:

```
conda activate myenv
```

5. To deactivate an active environment use:

```
conda deactivate my env
```

3.4 Creating a conda environment in the envs folder on the SD card with specifying the python version

Description: The following steps describe how to create new conda environments in the SD card. However, as mentioned in the section 3.1 and 3.2 descriptions, this only works when you create an environment without specifying the python version that is different from python 2.7. When I tried installing python version 3.8 I ran into an error.

1. Make sure you are in the correct directory where Miniforge is installed.

```
cd /home/ucsboep/miniforge-pypy3
```

2. Activate Miniforge: Activating the base environment:

```
source bin/activate
```

If this command succeeds, you should see `(base)` at the beginning of your terminal prompt.

3. Create a new conda environment

```
conda create --name segmentanything python=3.8
```

4. Proceed with instructions.

ERROR

Here is a snippet of the terminal error

Downloading and Extracting Packages

Preparing transaction: done Verifying transaction: done Executing transaction: done ERROR conda.core.link:_execute(945): An error occurred while installing package 'conda-forge::ca-certificates-2023.7.22-hcefe29a_0'. Rolling back transaction: done

[Errno 1] Operation not permitted: 'cacert.pem' -> '/media/ucsboep/NVIDIASD/envs/segmentanything/ssl/cert.pem' ()

(base) ucsboep@ucsboep-desktop:/media/ucsboep/NVIDIASD/envs$

Next Steps

Once again, I think this has to do with restricted permissions on the SD card. This seems to be a recurring issue. I'm afraid I'll continue to run into these issues whenever I try to install dependencies here. I will instead focus on removing unnecessary folders (or moving them to an SD card) on the NVIDIA machine and having conda environments stored in the internal storage instead.

4. Properly uninstalling Miniforge-pypy3 and Installing Miniforge 3

Description: Miniforge-pypy3 was initially installed on the machine, but I learned that miniforge-pypy3 is not as compatible with other dependencies as miniforge3. Therefore, I deleted miniforge-pypy3 and replaced it with miniforge3. Here's are the steps I took to uninstall miniforge from the Nvidia machine. I used miniforge documentation on github to create these steps. By doing this we can ensure we have a fresh start when installing a different conda version in our root:

4.1 Remove Shell Modifications

1. Open a terminal.
2. Navigate to the directory where Miniforge was installed, for example:

```
cd ~/miniforge-pypy3
```

3. Reverse the changes Miniforge made to your shell rc files:

```
./bin/conda init --reverse
```

4.2 Remove Miniforge Base Environment

1. In the same terminal, after reversing shell modifications, get the Miniforge base environment path:

```
CONDA_BASE_ENVIRONMENT=$(./bin/conda info --base)
```

2. Verify the path using:

```
echo $CONDA_BASE_ENVIRONMENT
```

3. If the path is correct, delete the Miniforge base environment directory:

```
rm -rf $CONDA_BASE_ENVIRONMENT
```

4.3 Remove Global Conda Configuration Files

1. Remove the `.condarc` file if it exists:

```
rm -f "${HOME}/.condarc"
```

2. Remove the `.conda` directory and its contents if they exist:

```
rm -fr ${HOME}/.conda
```

4.2 Restart Your Shell

Close your current shell and open a new one to ensure that all changes take effect.

By following these steps, you should be able to completely uninstall Miniforge from your system.

4.3 Installing Miniforge3 on your Machine

1. Navigate to Desired Directory: Open your terminal and use the `cd` (change directory) command to navigate to the directory where you want to download the Miniforge installer. In this example, let's navigate to the home directory:

```
cd /home/ucsboep
```

2. **Download Miniforge Installer:** Download the Miniforge installer script from the official Miniforge repository.

```
wget https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-Linux-x86_64.sh
```

3. Run Installer Script: Make the downloaded installer script executable and run it.

```
chmod +x Miniforge3-Linux-x86_64.sh
./Miniforge3-Linux-x86_64.sh
```

4. Accept License Agreement: During the installation process, read the license agreement and type 'yes' to accept it.

5. Choose Installation Location: Choose the installation location for Miniforge. The default location is suggested. Press Enter to confirm.

6. Initialize Conda: After installation, you'll be prompted if you want to initialize Miniforge by running `conda init`. Type 'yes' to proceed.

7. Update Shell Configuration: The initialization updates your shell's configuration files (e.g., `.bashrc`) to include the necessary paths for Conda.

8. Closing and Reopening Terminal: You can either close and reopen the terminal or manually execute the following command to activate Conda:

```
source ~/miniforge3/bin/activate
```

9. Confirmation and Usage: The installation process confirms success, and you now have Conda and Mamba installed on your machine!

5. Installing Segment Anything

Description: Although I was unable to use the SD card for storage purposes involving conda (described in sections 2 and 3), I moved some necessary files to the SD card (SCI_skunks-main and SCI_skunks-main-zip) and ended up having enough storage to create a conda environment for segment anything, and install required dependencies. I did not have enough storage, however, to install optional dependencies for mask post-processing, saving masks in COCO format, example notebooks, and exporting the model in ONNX format.

5.1 Creating an Environment for Segment Anything and Installing it

1. Create a Conda Environment:

```
conda create -n segment_anything_env python=3.8
```

Activate the environment:

```
conda activate segment_anything_env
```

2. Install Required Dependencies: Follow the instructions from the PyTorch website (https://pytorch.org/get-started/locally/) to install PyTorch and TorchVision with CUDA support

(recommended). The command I used was:

```
pip3 install torch torchvision torchaudio
```

3. Install Segment Anything: Once you have PyTorch and TorchVision installed, you can install Segment Anything using pip:

```
pip install git+https://github.com/facebookresearch/segment-anything.git
```

Alternatively, you can clone the repository and install it:

```
git clone git@github.com:facebookresearch/segment-anything.git
cd segment-anything
pip install -e .
```

4. Install Optional Dependencies: If you want to use the optional features of Segment Anything, such as mask post-processing, saving masks in COCO format, example notebooks, and exporting the model in ONNX format, you can install the following dependencies:

```
pip install opencv-python pycocotools matplotlib onnxruntime onnx jupyter
```

ERROR

At this point, I ran the command above with no issues until an error popped up and claimed I had no storage left. If the storage issue is resolved in the future the remaining instructions should work.

5. Run Example Notebooks (Optional): If you installed the optional dependencies and want to run the example notebooks, navigate to the directory where you cloned or installed Segment Anything and start Jupyter Notebook:

```
jupyter notebook
```

Remember to activate the environment whenever you want to work with Segment Anything:

```
conda activate segment_anything_env
```

Deactivate it when you're done

```
conda deactivate
```

Next Steps

Once the machine has more internal storage, the optional segment anything model dependencies can be installed. After this, masks can be generated and processed with ease!