[Making Simple Linux Distro from Scratch](https://www.youtube.com/watch?v=QlzoegSuIzg&t=149s) - video that walks through the compiling the kernel, creating the initramfs and init files. Tutorial uses syslinux to add to boot but this will not work on newer machines that do not support legacy boot. Instead, use GRUB

The following error may occur during compiling of the kernel:

---------------------------------------------------------------------

Make[1]:\*\*\*[scripts/Makefile.build:198:networking/tc.o] Error 1

Make:\*\*\*[Makefile:744:networking] Error 2

---------------------------------------------------------------------

To fix this you need to disable CONFIG\_TC in the .config file

CONFIG\_TC=y ---> CONFIG\_TC=n

Now move the kernel, init and initramfs to the /boot dir.

Add to GRUB via 40\_custom

Sudo vim /etc/grub.d/40\_custom - add the following below the comments:

Menuentry “MerlinOS” {

Insmod part\_gpt

Linux /boot/MerlinOS/kernel console=tty0 root=/dev/ram0

Initrd /boot/MerlinOS/init.gz

}

---------------------------------------------------------------------

Tty0 = default console

/dev/ram0 = root for OS (will need to create separate partition in the future.AT the moment this is using RAM for the root file-system)

Run sudo update-grub afterwards to update the GRUB config files

Add stuff to boot

Sudo nano /etc/systemd/system/check\_ethernet.service

[Unit]

Description=Check if there is an ethernet connection

After=network.target

[Service]

ExecStart=/usr/bin/python3 /home/merlin/Desktop/check\_ethernet.py

[Install]

WantedBy=multi-user.target

Sudo systemctl daemon-reload

Sudo systemctl enable check\_ethernet.service - enables service to run on boot

Sudo systemctl status check\_ethernet.service - check if everything is working; can see any errors.

CRC32- Sends crc32.sh to ACC, runs on terminal, returns crc32 value then deletes itself. Included in the crc32 for the ACC is the system info such as the kernel name, version and architecture, version of operating system, desktop environment and OS build, and all packages installed on the system. It does not factor in files.

#!/bin/bash

#Gets system info

SYSTEM\_INFO=$(uname -a)

#Returns OS info

OS\_INFO=$(cat /etc/os-release)

#Returns all of the installed packages

INSTALLED\_PACKAGES=$(dpkg -l)

#Creates one string with all the info

SYSTEM="$SYSTEM\_INFO $OS\_INFO $INSTALLED\_PACKAGES"

#Saves the string to a temporary file

echo -n "$SYSTEM" > /tmp/system\_info.txt

#Gets the crc32 of the temporary file

CRC32=$(crc32 /tmp/system\_info.txt)

#Returns the CRC32 to the console

echo "CRC32: $CRC32"

#Deletes the temporary file

rm /tmp/system\_info.txt

There could be problems getting a connection between the two computers via ethernet. Seems to work much better when both computers have a static ip. 192.168.1.2/24 for the maintenance and 192.168.1.3/24 255.255.255.00 for the ACC computer.

Set up laptop as a SSH server:

Sudo apt install openssh-server -y

Set up CRC32:

Sudo apt install libarchive-zip-perl

Data Loader Maintance Laptop ACC flow chart

With Ethernet connected

ACC software

CRC

Delete

Retrive

Send

ARINC Dataloader

Without Ethernet connected

ACC software

OS

OS

Maint. Laptop

ACC

ACC

Maint. Laptop

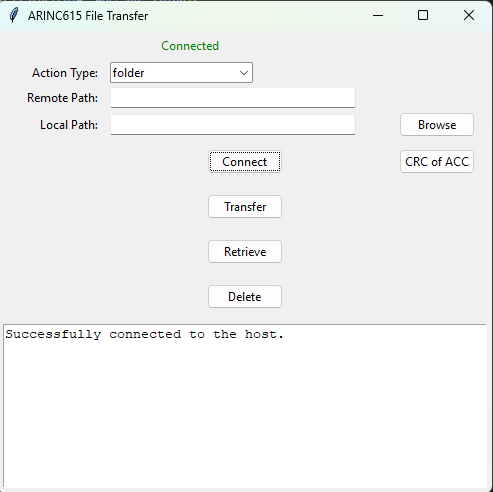
Steps to use the data loader.

Connect the maintenance laptop to the ACC with an Ethernet cable.

Start the Dataloader application on the maintenance laptop.

Press the ‘Connect’ button to establish a connection between the laptop and the ACC. The application will show a green ‘Connected’ verifying that the two computers are connected. A log will also be created in the information box following a successful connection

The main display looks as follows and allows for a range of options to interact with the ACC:



* Action type: Folder/File. Select the option that matches the type of interaction you want to have with the ACC
* Remote Path: This is the path on the ACC
* Local Path: This is the path on the maintanence laptop
* Browse button: This button opens a file dialog allowing the user to easily select the desired file, dir or path
* Transfer: Will transfer dir/files from the local path to the remote path
* Retrieve: Will retrieve dir/files from the remote path and place them at the local path
* Delete: Will delete dir/files from the remote path
* CRC of ACC: Returns the current CRC of the ACC (includes, OS & Hardware information alongside any installed packages). This is done by running the crc32.sh script on the ACC. This file is then deleted after running. DO NOT OPEN THE crc32.sh SCRIPT ON ANY LAPTOP THAT IS NOT RUNNING LINUX

During the transferring or retrieving of any dir/files, a crc32 is taken before the transfer and after the transfer. These two values are then compared to validate the transfer.

The information box will inform you of any errors that occur through out the transfer