**How to Install Vivado on Linux**

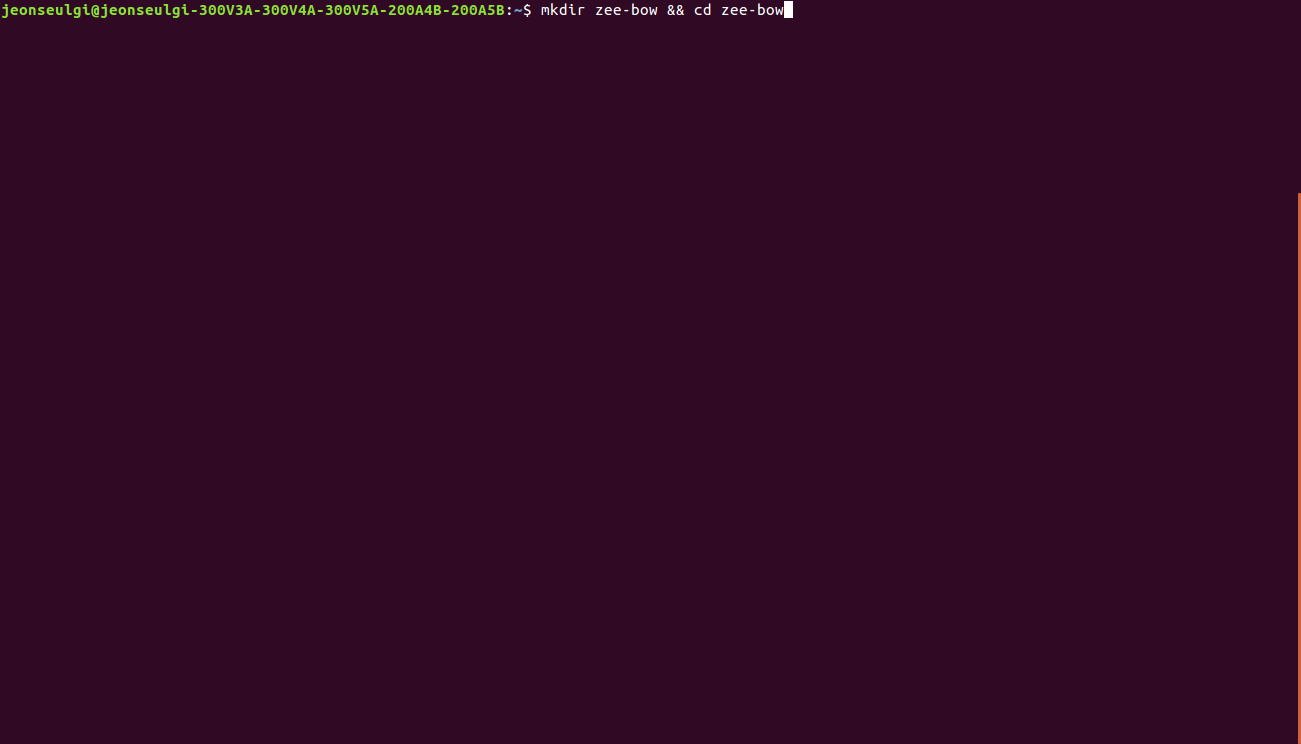
1. [**https://www.xilinx.com/support/download.html**](https://www.xilinx.com/support/download.html)이 사이트에 들어가서

**Vivado Design Suite – HLx Editions에** [Vivado HLx 2017.1: All OS installer Single-File Download](https://www.xilinx.com/member/forms/download/xef.html?filename=Xilinx_Vivado_SDK_2017.1_0415_1.tar.gz&akdm=1) 를 다운한다.

1. **Xilinx\_Vivado\_SDK\_2017.1\_04145\_1.tar.gz**를 다운받고 압축을 푼다.
2. 터미널을 킨다.
3. 터미널에 **sudo dpkg-reconfigure dash** 을 입력한다.
4. **No** 선택하고 엔터 (dash쉘에서 설치가 가능한 bash쉘로 변경됨)
5. 터미널 창을 하나 더 띄운다.
6. 터미널에 **./xsetup** 입력한다.
7. **Next** -> **3개다 I Agree 로 체크 후 Next** -> **vivado HL WebPACK 체크 후 Next** -> **SDK 체크 후 Next** -> home/컴퓨터에 지정한 이름(ex. koitt , jeonseulgi) 폴더로 지정 -> **Next**

## Step 1: Obtaining necessary files and repositories

1. mkdir로 zee-bow를 만들고 cd로 zee-bow를 들어간다.

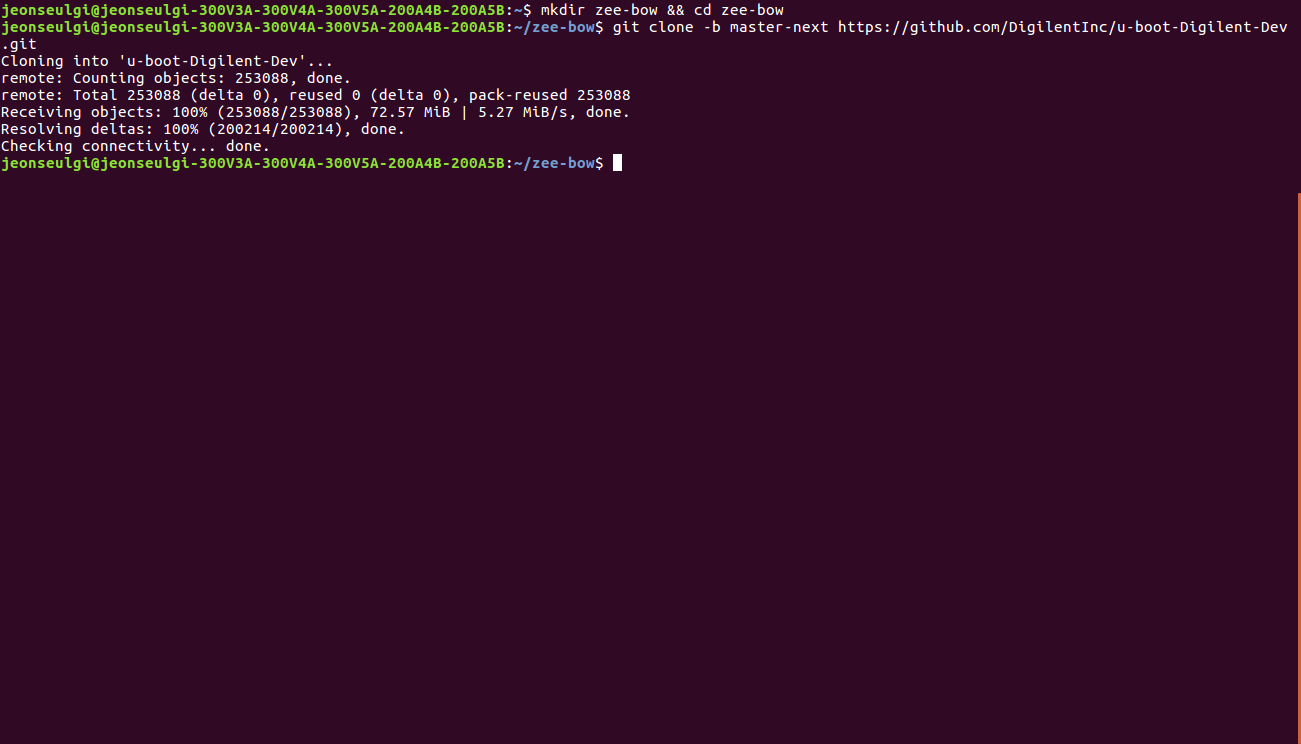


1. Git clone

[**https://github.com/DigilentInc/u-boot-Digilent-Dev**](https://github.com/DigilentInc/u-boot-Digilent-Dev)

에서 Clone or download눌러서 주소를 복사한 후

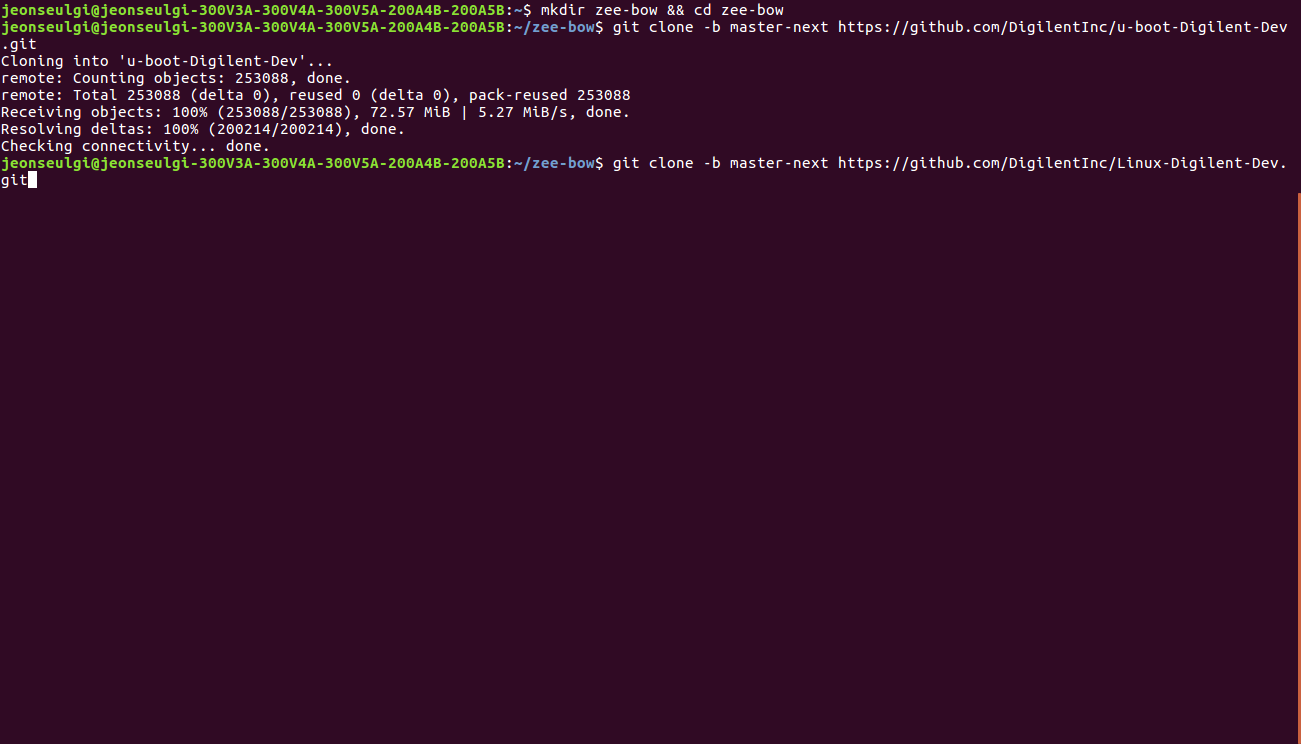
Git clone –b master-next 복사한 주소 붙여넣기

****

[**https://github.com/DigilentInc/Linux-Digilent-Dev**](https://github.com/DigilentInc/Linux-Digilent-Dev)

에서 Clone or download눌러서 주소를 복사한 후

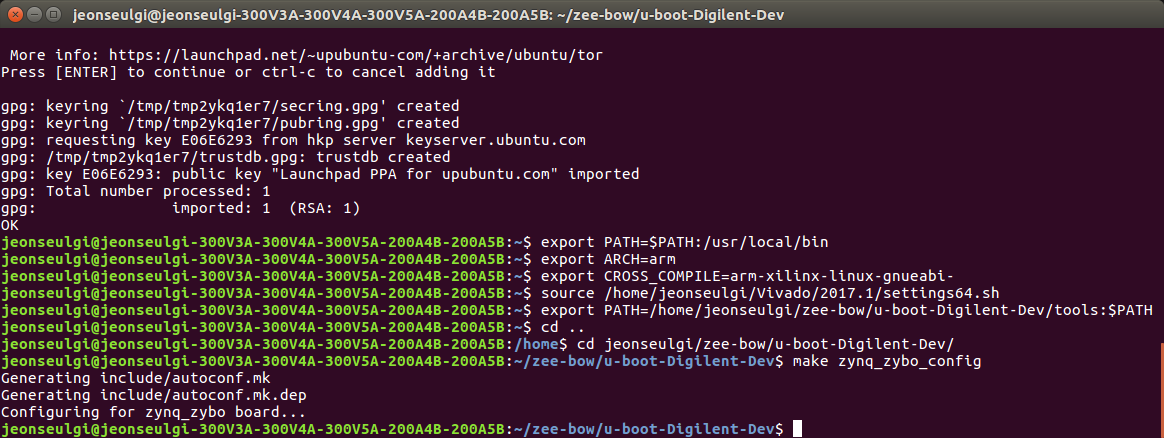
Git clone –b master-next 복사한 주소 붙여넣기

****

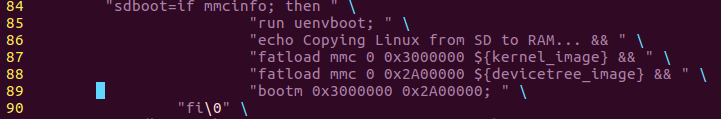
ls명령어를 통해 Linux-Digilent-Dev와 u-bbot-Digilent-Dev가 있는지 확인

## Step 2: Compiling U-Boot

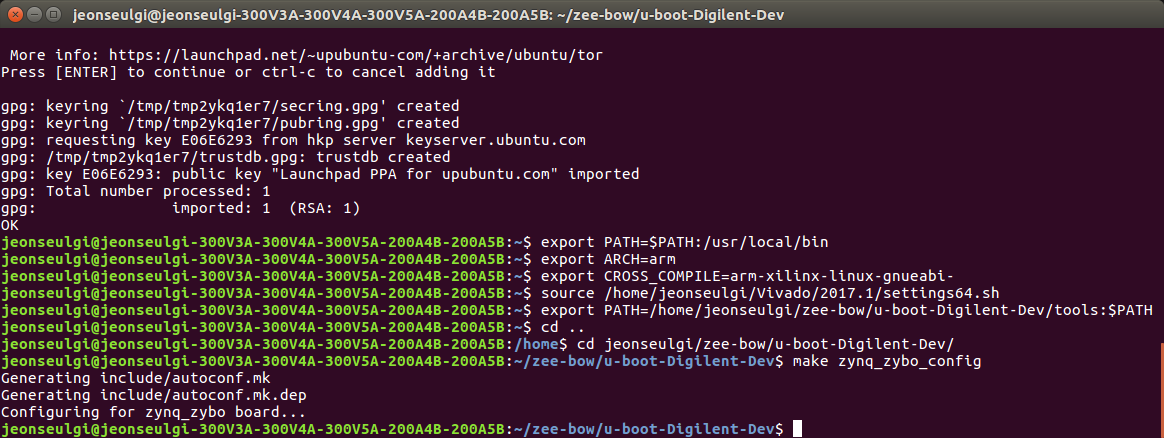
환경변수 설정



zee-bow/u-boot-Digilent-Dev/include/configs안에서 vi zynq\_zybo.h한 후 밑에 소스처럼 변경한다.



make zynq\_zybo\_config를 해준다.



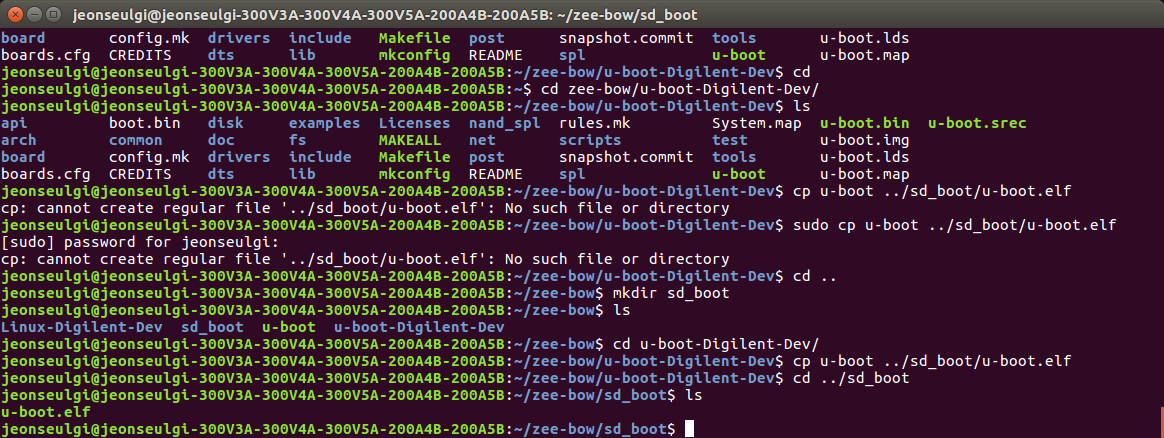
만약 환경변수 설정을 잘못하게 될 경우

the command could not be located because '/bin' is not included in the PATH environment variable에러가 뜬다.

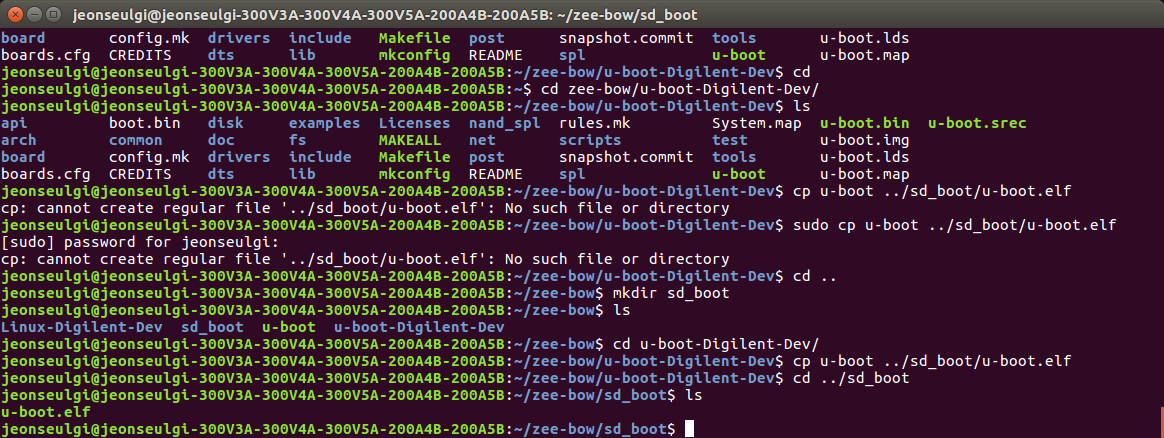
make를 한다

C:\Users\KOITT\Desktop\3.png

ls명령어를 통해서 u-boot에 관련된 목록들이 만들어 진다.



mkdir sd\_boot 한 후에 cp u-boot ../sd\_boot/u-boot.elf한다. 그리고 잘 cp됐는지 ls명령어로 확인한다.

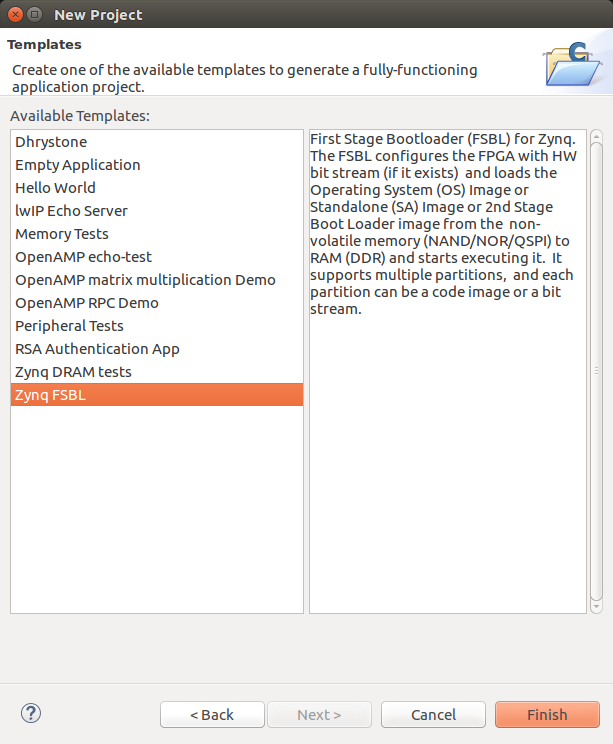
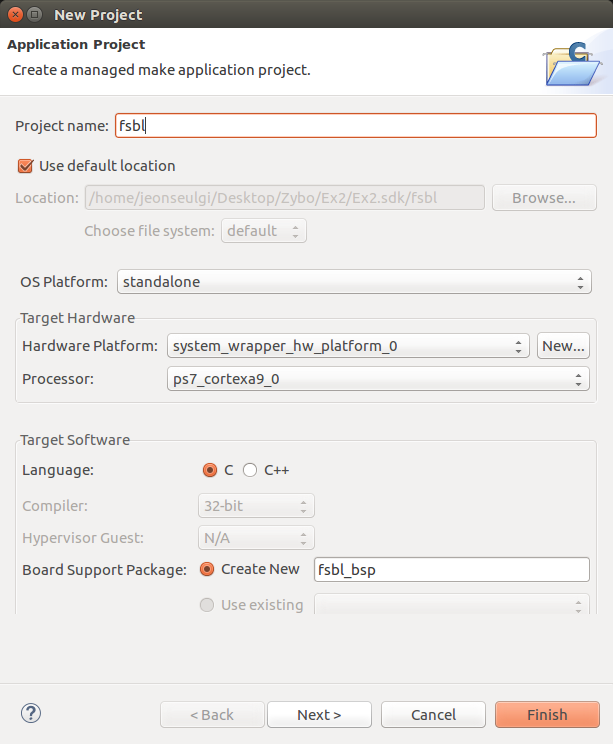


Step 3: Building the base design

Embedded System Design on Zynq 책에서 p.59~p.82까지 실행한 후

Application project를 생성한다.

Step 4: First Stage Boot Loader



## Step 5: Building BOOT.bin

상단에 Xilinx Tools에서 Create Boot Image를 누른다.

Output BIF file path: /home/jeonseulgi/zee-bow/sd\_boot/output.bif로 변경하면 맨 밑에 Output path가 자동적으로 /home/jeonseulgi/zee-bow/sd\_boot/BOOT.bin으로 변경된다.

그리고 boot image partitions에 File path에서 fsbl.elf와 wrapper.bit와 u-boot.elf가 있는지 확인한다. 없으면 추가 해준다.

**fsbl.elf**

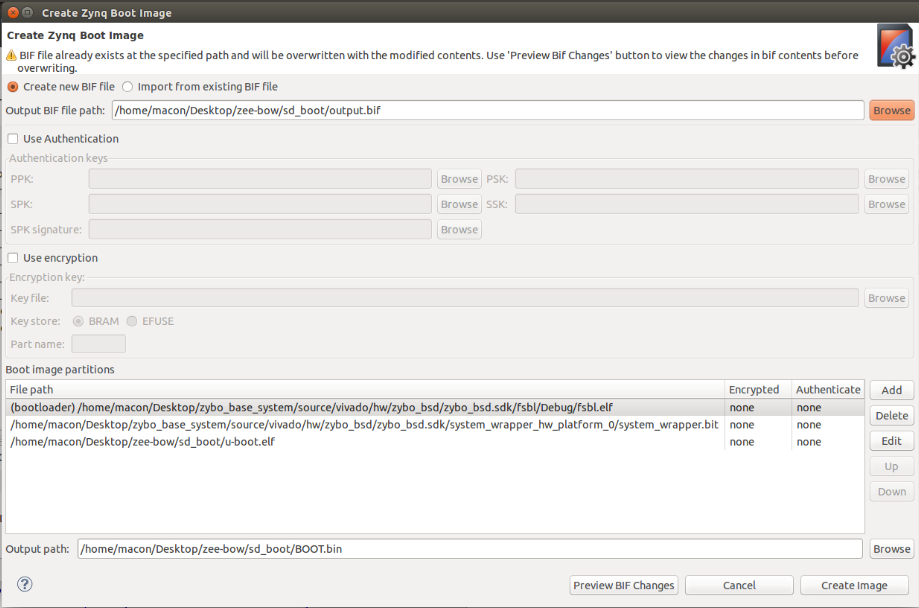
zybo\_base\_system/source/vivado/hw/zybo\_bsd/zybo\_bsd.sdk/fsbl/debug folder

**wrapper.bit**

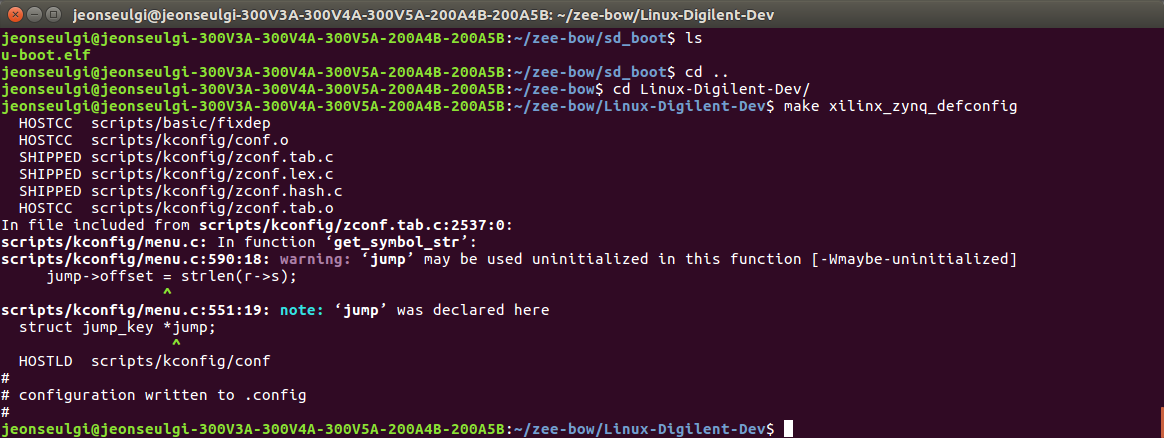
zybo\_base\_system/source/vivado/hw/zybo\_bsd/zybo\_bsd.sdk/system\_wrapper\_hw\_platform\_0 folder

**u-boot.elf**

zee-bow/sd\_boot folder 밑에는 예시(깜박하고 캡쳐안함)



## Step 6: Building the Linux kernel



**Step 7: Generating the Device Tree Blob**

Vi zynq-zybo.dts로 들어가서 42(밑에사진에서44번처럼 변경), 51(53번처럼), 60줄(62번처럼)을 수정한다.

