

Want to play a game?

Gotchas & Solutions

Compiling **hal.c**

1. Is the Makefile named EXACTLY **Makefile**?
2. Are you sure you have **make** installed?
 - a. `sudo apt install make`
3. Make sure you have **libelf-dev** installed:
 - a. `sudo apt install libelf-dev`

Still getting errors using Makefile?

1. `obj-m += hal.o` needs to match the name of your **C** file. Meaning that for simplicity make sure you named your **C** file exactly **hal.o**
2. In the Makefile that has the following code:

```
obj-m += hal.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

Try changes the `M=$(PWD)` to `M=$(shell pwd)`

```
obj-m += hal.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=$(shell pwd) modules
clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(shell pwd) clean
```

Error about: “/arch/x86/syscalls/syscall_32.tbl”

1. If you have not done a **sudo apt update** now is the time
2. Follow this [guide here](#)
3. Pray to the kernel gods that it works

Compiling **skynet.c** & **joshua.c**

1. Make sure the Makefile is in the same folder as your **skynet.c** or **joshua.c** file.
2. Is the Makefile named EXACTLY **Makefile**?
3. The first line: **obj-m += ...** should match the name of your **C** file. So it should either be:
 - a. **obj-m += skynet.o**
 - b. **obj-m += joshua.o**

Comparing your syntax with Phil

If nothing seems to work, you can always compare your code with the mighty Phil's examples, displayed on his [Github](#).

If you find Phil's code helpful, don't forget to leave him a star on Github :)