For all of this you require:

nasm, qemu (added to path)

pyosdevkit.py in the same directory as python file you’ll develop your operating system

Blank Operating System:

1. import pyosdevkit to your file by:

from pyosdevkit import \*

2. specify output asm file name (name have .asm at the end) by:

primary\_file\_asm(“example.asm”)

3. tell translator to jump to current memory address (make loop) by:

pta.jump.current()

4. tell translator that file will be bootable by:

pta.make\_bootable()

5. tell pyosdevkit to end asm file entry by:

primary\_asm\_output\_write()

6. compile and execute operating system (requires nasm and qemu in the path) by:

run()

7. After all this steps file should look like this:

from pyosdevkit import \*

primary\_file\_asm(“example.asm”)

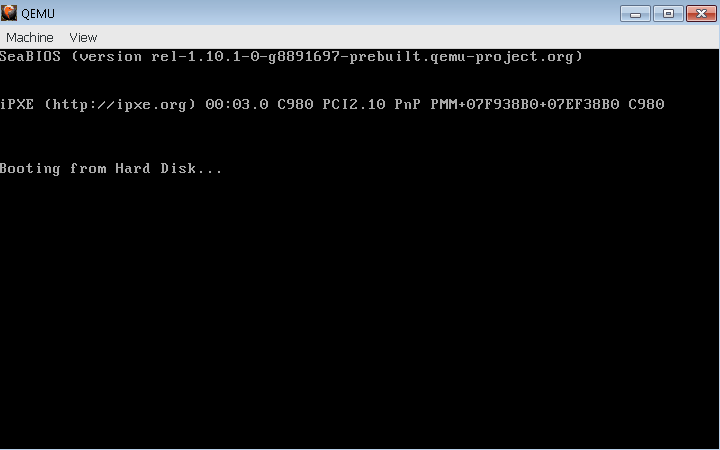
pta.jump.current()

pta.make\_bootable()

primaryasm\_output\_write()

run()

Final Result:



Adding text to the screen:

1. Setup blank operating system by copping this code to your file:

from pyosdevkit import \*

primary\_file\_asm(“example.asm”)

pta.jump.current()

pta.make\_bootable()

primary\_asm\_output\_write()

run()

2. All code between primary\_file\_asm(“example.asm”) and pta.jump.current() will be executed.

3. Write string using bios by:

pta.bios.write\_string(“Hello PyOSDevKit!”)

4. All is done your code should look like this:

from pyosdevkit import \*

primary\_file\_asm(“example.asm”)

pta.bios.write\_string(“Hello PyOSDevKit!”)

pta.jump.current()

pta.make\_bootable()

primary\_asm\_output\_write()

run()

Final Result:

