Guide to using PintOS

Prof: Jorge Gonzalez
jgonzalez@utec.edu.pe
TA: Martin Carrasco
martin.carrasco@utec.edu.pe

March 31, 2020

1 Making a docker container for PintOS

1. Pull the docker image from Docker Hub
2. Create a volume to have data be persistant e.g. *sudo docker volume create my_volume PATH* (see additional info in `item 5`).
3. Create the Dockerfile *(Use the template Dockerfile included to guide you, notice that in our template we already pull the image).*
   - Install dependencies
   - Set enviromental variables
4. Build the container e.g. *sudo docker build -t pintos .*
5. Run the container e.g. *sudo docker run -it –volume my_volume:/pint-os –name pint-sim pintos*. For more details about using volumes, please refer to Docker Documentation.

2 Seting up and compiling PintOS

1. Make sure all the dependencies were installed correctly
2. Compile the following submodules

- `userprog`
- `vm`
- `filesys`

3. Edit **src/utils/Makefile** to replace `LDFLAGS= -lm` to `LDLIBS = -lm` then compile **src/utils**
4. Edit **src/thread/Make.vars** and change `SIMULATOR=` to `SIMULATOR= -qemu` then compile **src/threads**
5. Change **src/utils/pintos** `$sim=bochos` to `$sim=qemu`
6. Change **src/utils/pintos**, check `$name = find_file('kernel.bin')` to point to **threads/build/kernel.bin**
7. Change **src/utils/pintos** `my (@cmd) = ('qemu')` to `my (@cmd) =qemu-system-x86_64`
8. Edit in **src/utils/Pintos.pm**, `$name = find_file('loader.bin')` and point it to **threads/build/loader.bin**
9. See PintOS Documentation for compile and run. Notice that we are using Qemu.