

Getting started in INF113

In this course, we will be using the Linux operating system. We recommend that you use a virtual machine with Ubuntu. This tutorial will guide you through the process, from setting up the virtual environment to running the first code samples from the book.

1. [Download VirtualBox and Ubuntu](#)
2. [Create a virtual machine](#)
3. [Installing Ubuntu on your VM](#)
4. [VirtualBox Guest Additions](#)
5. [Setup for simulators and homework](#)

Download VirtualBox and Ubuntu

Use the following links to download Ubuntu and install VirtualBox.

[VirtualBox](#)

[Ubuntu](#)

For VirtualBox, select the large blue download button and then select your current operating system. For Ubuntu, simply click the download button. Note that Ubuntu is just an iso file that we will use later.



The screenshot shows the VirtualBox website. On the left is a sidebar with links: About, Screenshots, Downloads, Documentation (with sub-links for End-user docs and Technical docs), Contribute, and Community. The main content area has a large 'VirtualBox' logo and a 'Download VirtualBox' section. Below this, it says 'Here you will find links to VirtualBox binaries and its source code.' The 'VirtualBox binaries' section includes a disclaimer about the license and links to the latest builds for version 6.0 (supported until July 2020) and version 5.2 (supported until July 2020). A 'VirtualBox 6.1.22 platform packages' section contains a list of links for Windows hosts, OS X hosts, Linux distributions, Solaris hosts, and Solaris 11 IPS hosts, with a large left-pointing arrow next to it. Below this, it states that binaries are released under the GPL version 2, provides a link to the changelog, and mentions that SHA256 checksums should be favored over MD5 checksums. A note at the bottom recommends upgrading guest additions after upgrading VirtualBox.

VirtualBox
Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.0 packages, see [VirtualBox 6.0 builds](#). Please also use version 6.0 if you need to run VMs w supported until July 2020.

If you're looking for the latest VirtualBox 5.2 packages, see [VirtualBox 5.2 builds](#). Please also use version 5.2 if you still need support until July 2020.

VirtualBox 6.1.22 platform packages

- [Windows hosts](#)
- [OS X hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)
- [Solaris 11 IPS hosts](#)

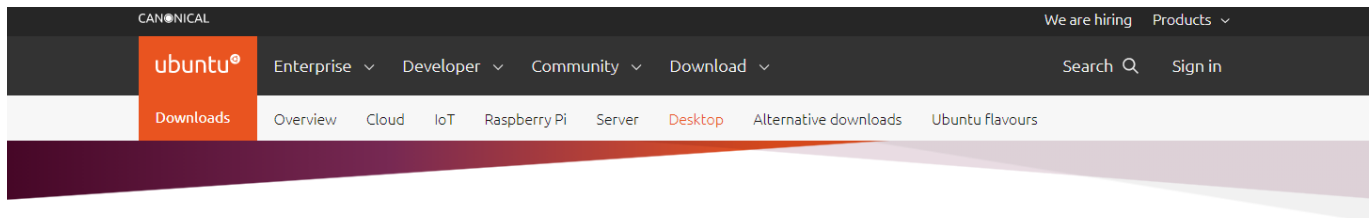
The binaries are released under the terms of the GPL version 2.

See the [changelog](#) for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored*

- [SHA256 checksums](#), [MD5 checksums](#)

Note: After upgrading VirtualBox it is recommended to upgrade the guest additions as well.



Download Ubuntu Desktop

Ubuntu 20.04.2.0 LTS

Download the latest [LTS](#) version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years, until April 2025, of free security and maintenance updates, guaranteed.

[Ubuntu 20.04 LTS release notes](#)

Recommended system requirements:

- ✓ 2 GHz dual core processor or better
- ✓ 4 GB system memory
- ✓ 25 GB of free hard drive space
- ✓ Internet access is helpful
- ✓ Either a DVD drive or a USB port for the installer media

Download

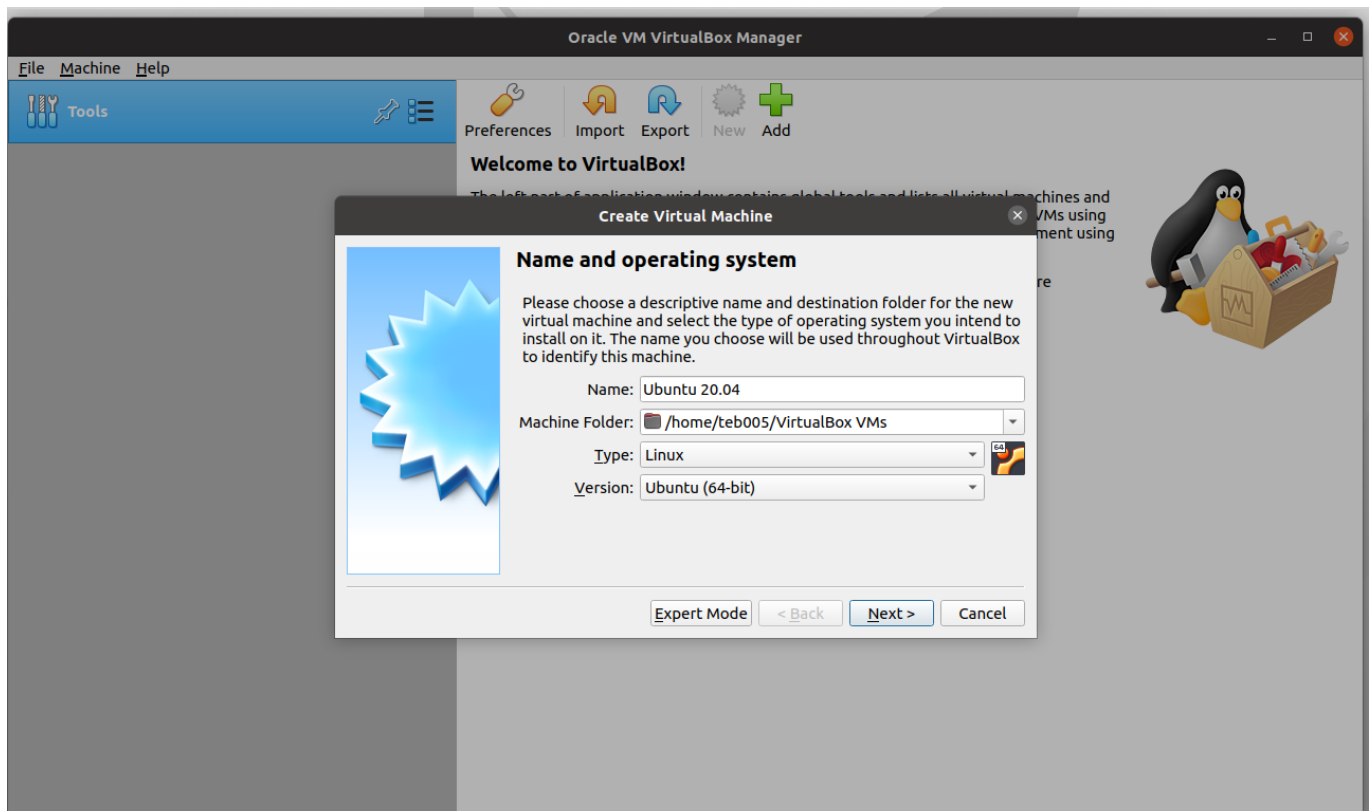


For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors, and past releases [see our alternative downloads](#).

Create a virtual machine

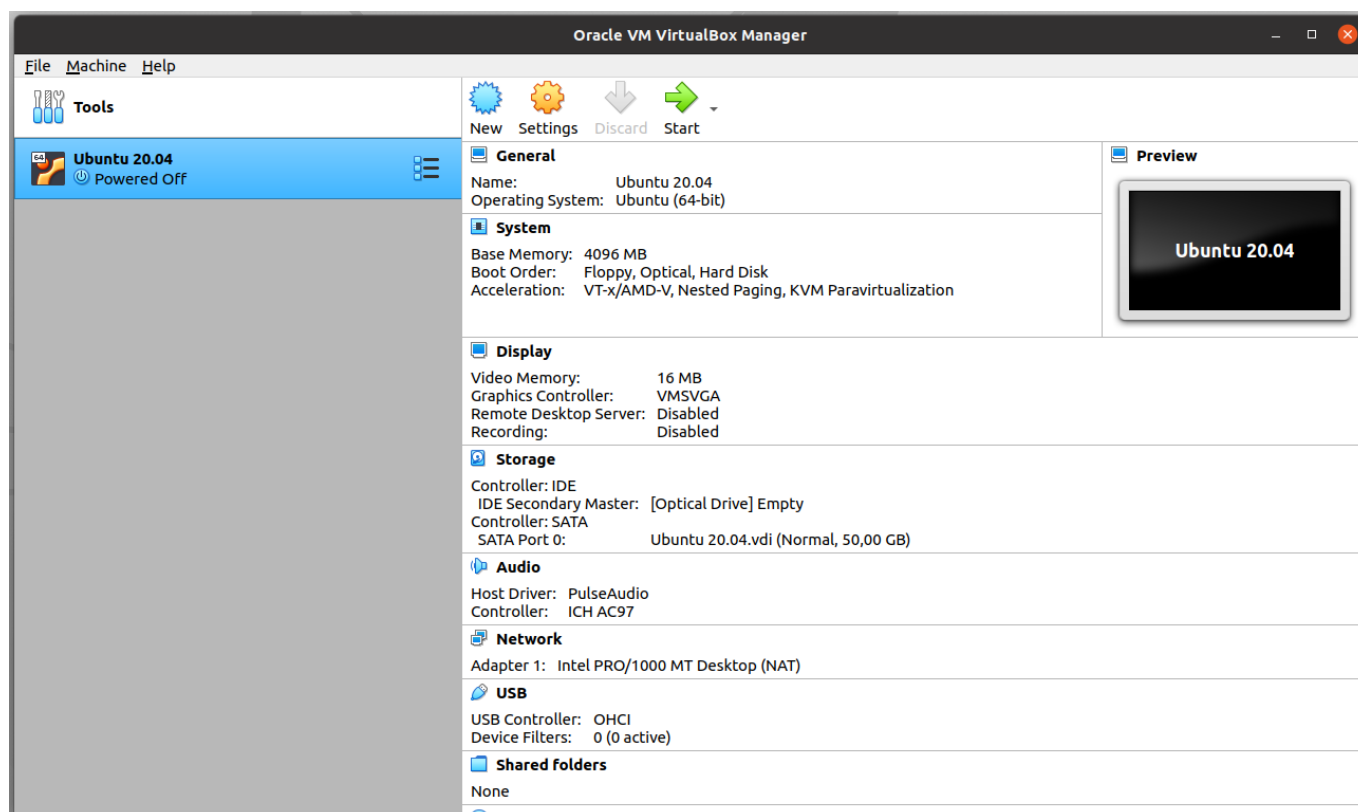
Once you have installed VirtualBox and downloaded Ubuntu, you can create a new virtual machine (VM). The following steps will guide you through the process.

1. Start by running VirtualBox and then select **new**. Give the VM a name and select Linux and Ubuntu (64-bit). It should look something like this.



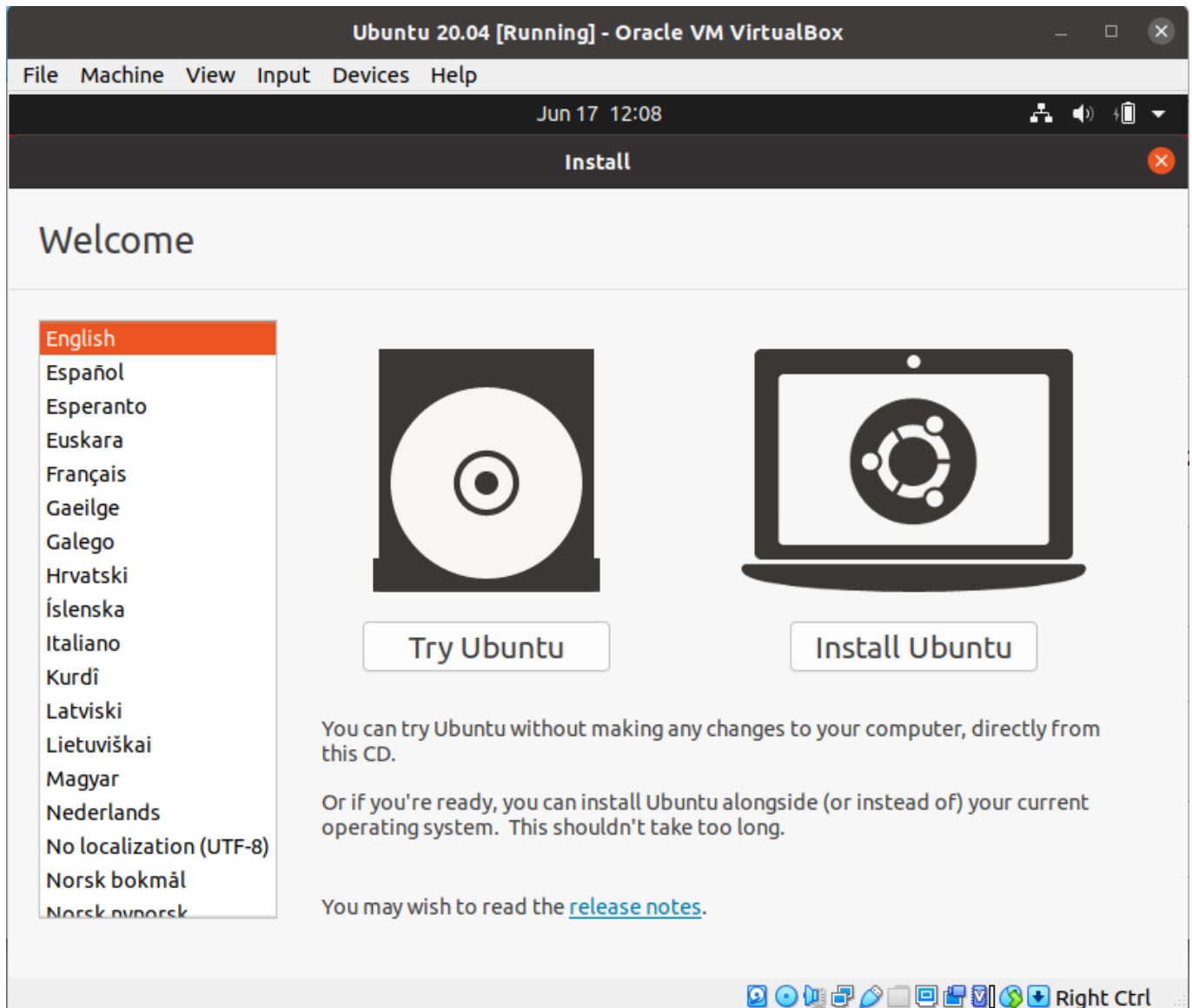
2. Click Next and select the amount of memory to be allocated to the virtual machine. It is recommended to use a minimum of **2 GB**, but you can use more depending on your setup. This option can be changed later, so you can experiment with different values.
3. Leave the next option on the default **Create a virtual hard disk now** and click **create**.
4. Continue with the default option **VDI**.
5. Either option is fine here, but we recommend using the **Fixed size** option for better performance, especially if you use an older machine.
6. Now, you need to select the size of the virtual hard disk. A minimum of 10 GB is recommended, but again, you can set it higher if you have more storage available. Unlike the amount of memory, the disk size can not easily be changed after creating the VM.

The new VM should now appear in the list on the left side of the VM manager, like this.



Installing Ubuntu on your VM

1. First, we need to add the Ubuntu iso file we downloaded earlier. Click on **Tools** from the left sidebar, select **Media** (through the small bulletin icon), and then **Optical disks**. From here, select **Add** and navigate to the Ubuntu iso file.
2. Select your new VM from the left sidebar and click start. This will prompt a **Select start-up disk** window. If you did the previous step correctly, Ubuntu should now be in the dropdown menu. Select Ubuntu and click **Start**.
3. After some time, Ubuntu should start, and an **install** window will open. It should look like this.



You will not be able to resize the VM window yet, but don't worry; we will fix that later.

4. Select **Install Ubuntu** and then choose your keyboard layout.
5. Leave it on **Normal installation** and continue.
6. Again, leave it on the default **Erase disk and install Ubuntu** and click **Install Now** (you will get a warning but just continue).
7. Choose location if you want and then select username and password.
8. After the installation is finished, you will be prompted with a restart message. Click **Restart Now** and then press Enter when it asks you to remove the installation medium.
9. When the system boots back up, skip through the questions until you are at the desktop.

That's it! You have now completed the Ubuntu installation, but we still need a few more things before you can start working. The first quality of life change is to make the VM window resizable and allow for bidirectional clipboard and drag and drop.

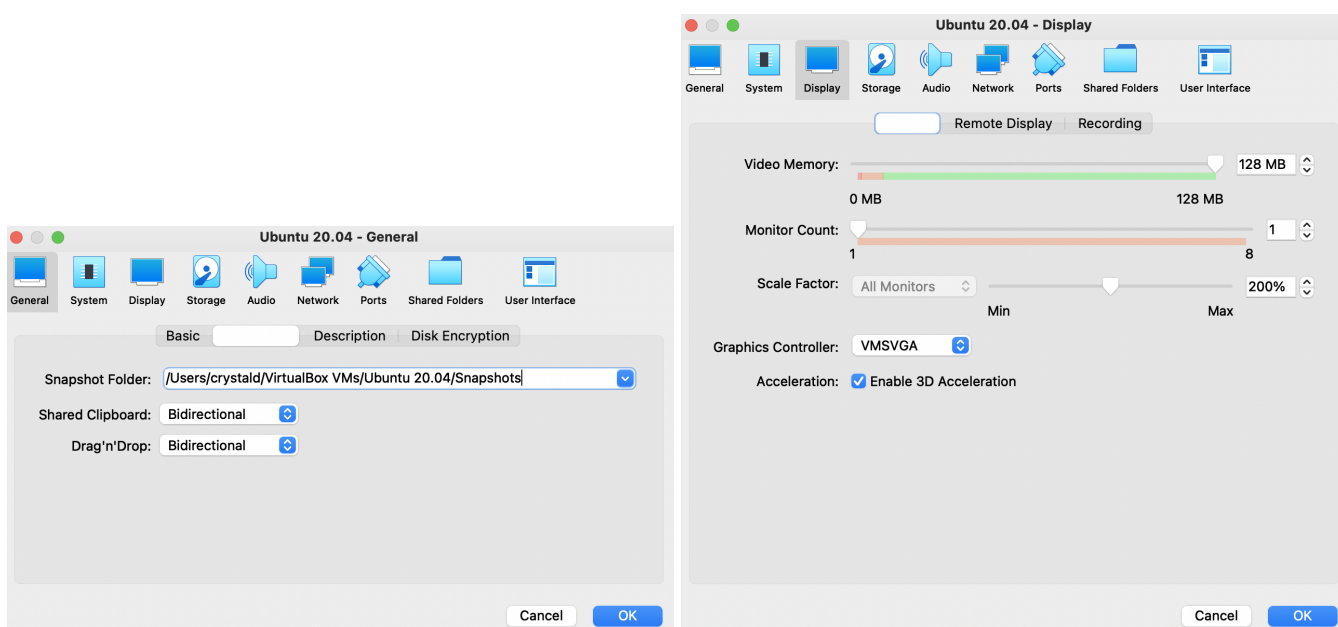
VirtualBox Guest Additions

1. From the Ubuntu desktop, select the icon in the bottom left corner (Show Applications) and open up a terminal. Then type the following command (you will have to provide your password and verify that you want to continue)

```
sudo apt install linux-headers-$(uname -r) build-essential dkms
```

2. When it is done, go to the **Devices** tab on the VM window and select **Insert Guest Additions CD image....** (If you get a popup saying "Could not find the guest additions disk image file," simply select the **Download** option.)
3. When the guest additions CD image has been inserted, you will get a prompt in Ubuntu asking if you would like to run it; select **Run**. When the installation is finished, restart the VM.

Now, when you resize the window, the resolution in the VM should also change accordingly. To fix the clipboard, enter the VM settings, either through machine -> settings on the VM window or the settings icon in the VM manager. Under **General/Advanced**, you can change the shared clipboard and drag and drop settings. While we are here, you might also be interested in the **System** and **Display** tabs. Here you can change the amount of memory, CPU cores, execution cap, and video memory. Especially video memory is recommended to increase to the max for a less laggy experience. Note that you can not change most of this while the VM is running.



Setup for simulators and homework

There is a lot of code in the book, and most of the homework relies on code written by the book's authors. This part of the installation guide will help you clone the necessary code and give a recommendation on how to write and run code in the virtual machine.

First, start by installing Git, GCC, and Python on the VM. This is as simple as running the following commands in a terminal.

```
sudo apt update
```

```
sudo apt install build-essential
```

```
sudo apt install git
```

```
sudo apt install python-is-python3
```

You can then verify that everything is working by testing the following commands.

```
python --version
```

```
git clone https://github.com/remzi-arpacidusseau/ostep-homework.git
```

make

```
./cpu "A"
```

```
Open [v] [f] cpu.c ~/Documents/ostep-code/intro Save [≡] - □ ×

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "common.h"
4
5 int main(int argc, char *argv[])
6 {
7     if (argc != 2) {
8         fprintf(stderr, "usage: cpu <string>\n");
9         exit(1);
10    }
11    char *str = argv[1];
12
13    while (1) {
14        printf("%s\n", str);
15        spin(1);
16    }
17    return 0;
18 }
19
```

```
kenneth@kenneth-VirtualBox: ~/Documents/ostep-code/Intro$ ls
ostep-code  ostep-homework
kenneth@kenneth-VirtualBox:~/Documents$ cd ostep-code/
kenneth@kenneth-VirtualBox:~/Documents/ostep-code$ ls
cpu-api      include      threads-api  threads-intro  vm-intro
cpu-sched-lottery  intro      threads-bugs  threads-locks
dist-intro    README.md   threads-cv   threads-sema
kenneth@kenneth-VirtualBox:~/Documents/ostep-code$ cd intro/
kenneth@kenneth-VirtualBox:~/Documents/ostep-code/intro$ ls
common.h  common_threads.h  cpu.c  io.c  Makefile  men.c  README.md  threads.c
kenneth@kenneth-VirtualBox:~/Documents/ostep-code/intro$ gedit cpu.c &
[1] 6838
kenneth@kenneth-VirtualBox:~/Documents/ostep-code/intro$ make
gcc -o cpu cpu.c -Wall
gcc -o men men.c -Wall
gcc -o threads threads.c -Wall -pthread
gcc -o io io.c -Wall
kenneth@kenneth-VirtualBox:~/Documents/ostep-code/intro$ ./cpu "A"
A
A
A
A
A
A
A
kenneth@kenneth-VirtualBox:~/Documents/ostep-code/intro$
```

6 / 7

ostep-homework/cpu-intro folder (**cd ..** takes you one folder out). Then test the following command.

```
./process-run.py -l 3:0 -L 5 -c
```

If everything worked, the output should look like this.

Time	PID: 0	CPU	IOs
1	RUN:io	1	
2	WAITING		1
3	WAITING		1
4	WAITING		1
5	WAITING		1
6	WAITING		1
7*	RUN:io_done	1	
8	RUN:io	1	
9	WAITING		1
10	WAITING		1
11	WAITING		1
12	WAITING		1
13	WAITING		1
14*	RUN:io_done	1	
15	RUN:io	1	
16	WAITING		1
17	WAITING		1
18	WAITING		1
19	WAITING		1
20	WAITING		1
21*	RUN:io_done	1	