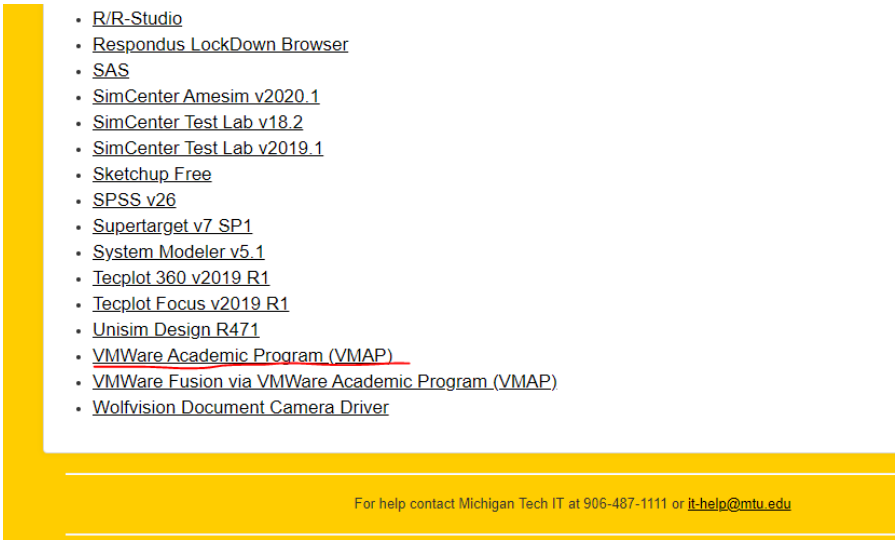


How to Install VM and ThreadMentor

In order to test your programs of CS3331 with ThreadMentor, one way is to install a Virtual Machine (VM) running Red Hat with ThreadMentor, and test your code within this VM.

1. Download VMWare from MTU Software Distribution Center.

- a) Login to MTU Software Distribution Center: <https://downloads.it.mtu.edu/>
- b) Scroll down to find **VMWare Academic Program (VMAP)** if you are using a Windows PC, and **VMWare Fusion via VMWare Academic Program (VMAP)** if you are using a MAC. Please note that for the time being, Fusion is not compatible with M1 chip.




- [R/R-Studio](#)
- [Respondus LockDown Browser](#)
- [SAS](#)
- [SimCenter Amesim v2020.1](#)
- [SimCenter Test Lab v18.2](#)
- [SimCenter Test Lab v2019.1](#)
- [Sketchup Free](#)
- [SPSS v26](#)
- [Supertarget v7 SP1](#)
- [System Modeler v5.1](#)
- [Tecplot 360 v2019 R1](#)
- [Tecplot Focus v2019 R1](#)
- [Unisim Design R471](#)
- [VMWare Academic Program \(VMAP\)](#)
- [VMWare Fusion via VMWare Academic Program \(VMAP\)](#)
- [Wolfvision Document Camera Driver](#)

For help contact Michigan Tech IT at 906-487-1111 or it-help@mtu.edu

- c) This link will lead you to the software description, the VMAP will provide a 1-year license without any charge. In case it expires in 1 year, just come again to download, it will provide another free of charge 1-year license. Click the link **VMAP System**.

VMWare Academic Program (VMAP)

Name	VMWare Academic Program	
Software Included	Some of the software available is: <ul style="list-style-type: none">• VMWare Fusion• VMWare Workstation• VMWare vCenter Server Standard• VMWare vCloud Suite Standard• VMWare vSphere	
Vendor	VMware	
Platforms	Windows, Mac OS X, Linux	
Expiration	Licenses for products are valid for 12 months starting with the 1st of the month the product was downloaded. Product may be re-licensed 12 months after the initial download.	
Description	Michigan Tech has a Campus VMAP Subscription.	
Distribution	All media and licenses are handled by the VMAP system hosted by OnTheHub.	
Restrictions	Software from this program has very specific guidelines on which usage is allowed (and not allowed). Please see the VMWare Academic Licensing Overview page to ensure you are compliant with VMWare's usage policy.	
URL	Follow this link to access the VMAP System . Login using your Michigan Tech ISO userid and password.	
Support Information	VMWare does NOT provide technical support for any products licensed under the VMAP.	
Important Notes	Software obtained under by you under this program may only be installed on your own personally-owned equipment. The program does provide licenses for university owned equipment used in research and instruction (including Senior Design). If you need licenses for these purposes, please contact IT User Services.	

For help contact Michigan Tech IT at 906-487-1111 or it-help@mtu.edu

- d) The **VMAP System link** will lead you to a Product Search webpage, search for **VMware**, and choose **VMware Workstation 16.x Pro** if you are using a Windows PC, **VMware Fusion 12.x Pro** if you are using a MAC.

VMware vSphere 6.x Enterprise Plus ▾

VMware vSphere 7.x Enterprise Plus ▾

VMware Workstation 14.x Pro ▾

VMware Workstation 15.x Player ▾

VMware Workstation 15.x Pro ▾

VMware Workstation 16.x Player ▾

VMware Workstation 16.x Pro ▾

VMware eLearning ▾

VMware vSphere Install Configure and Manage 6 Labs ▾

- e) Add it to cart, then check out to download.
f) Install the VMware software as default settings.

2. Install Red Hat Enterprise Linux 8

Download Red Hat ISO from the official website, you will need to register on the website, remember your username and password, you will need this again when enabling Red Hat repositories.

<https://developers.redhat.com/products/rhel/download>.

3. Create the Red Hat VM

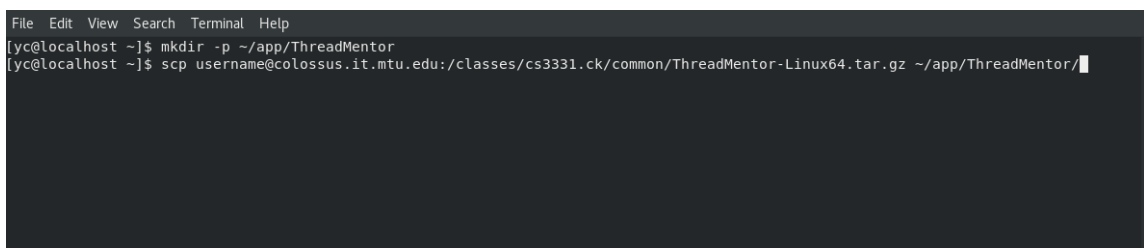
- a) Open VMware and Pick **File** -> **New Virtual Machine**.
- b) Choose Typical VM Configuration.
- c) Specify the path to your downloaded Red Hat iso image.
- d) Setup your Linux name, username and password.
- e) Specify the VM name and folder path.
- f) Specify disk space to the disk image file about to be created. This image file will host the operating system and everything in it. Click Finish to create your VM.
- g) Later you can reconfigure how many processors you want VMWare to emulate, how many cores per processor should you use, and amount of RAM memory for your virtual machine.

4. Install ThreadMentor.

- a) Start your newly installed VM.
- b) Create directory for ThreadMentor.
 - i. Open a terminal
 - ii. Create ThreadMentor directory. You can choose any directory you like, just remember to modify the path in your Makefile later to compile your code.
This example creates folder path of ~/app/ThreadMentor, type:
mkdir -p ~/app/ThreadMentor
 - iii. Download ThreadMentor from CS3331 common directory, type:

```
scp username@colossus.it.mtu.edu:/classes/cs3331.ck/common/ThreadMentor-Linux64.tar.gz  
~/app/ThreadMentor/
```

Remember to use your own MTU username to replace `username` in the command.
Type in your MTU password to download ThreadMentor file.



```
File Edit View Search Terminal Help  
[yc@localhost ~]$ mkdir -p ~/app/ThreadMentor  
[yc@localhost ~]$ scp username@colossus.it.mtu.edu:/classes/cs3331.ck/common/ThreadMentor-Linux64.tar.gz ~/app/ThreadMentor/
```

- iv. Untar the file into your ThreadMentor folder, type:
tar -xvf ~/app/ThreadMentor/ThreadMentor-Linux64.tar.gz -C ~/app/ThreadMentor/

```
File Edit View Search Terminal Help
[yc@localhost ~]$ tar -xvf ~/app/ThreadMentor/ThreadMentor-Linux64.tar.gz -C ~/app/ThreadMentor/
```

c) Add the visualization system to the execution path by adding the ThreadMentor bin directory to ~/.bashrc file

- i. Type **vim ~/.bashrc**
- ii. Type **i** to change vim into insert mode
- iii. Add your ThreadMentor bin directory to the last line:
export PATH= \$PATH:~/app/ThreadMentor/bin
- iv. Press **Esc** to exit insert mode.
- v. Press **:wq** and hit **Enter** to save and exit vim
- vi. Logout and back in your VM again to let configuration be active.

d) Install basic development tools.

i. Enable sudo access on your Red Hat:

Type: **su** , then input your Red Hat root password

Type: **usermod -aG wheel RedHatUserName**, remember to replace RedHatUserName with your own Red Hat VM username.

Log back in your Red Hat VM username, type: **su RedHatUserName**, remember to replace RedHatUserName with your own Red Hat VM username.

```
[yc@localhost ~]$ su
Password:
[root@localhost yc]# usermod -aG wheel yc
[root@localhost yc]# su yc
[yc@localhost ~]$
```

ii. Enable Red Hat repositories:

Type: **sudo subscription-manager register** , this will need your Red Hat username and password used for downloading Red Hat ISO image.

Type: **sudo subscription-manager refresh**

Type: **sudo subscription-manager attach --auto**

iii. Install Development Tools

Type: **sudo yum groupinstall "Development Tools"**

e) Install gtk+ (shared library support for Threadmentor visualization).

i. Enable EPEL 7 repository that contains gtk+.

Type: `sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm`

ii. Install gtk+.

Type: `sudo yum install gtk+`

5. Test code with ThreadMentor Visual.

a) Configure IFLAGS, TMLIB and TMLIB_NV in your Makefile according to your local ThreadMentor directory. This is an example.

```
1 CC = c++
2 FLAGS =
3 CFLAGS = -g -O2 -Wno-write-strings -Wno-cpp
4 DFLAGS = -DPACKAGE=\"threadsystem\" -DVERSION=\"1.0\" -DPTHREAD=1 -DUNIX_MSG_Q=1 -DSTDC_HEADERS=1
5 IFLAGS = -I/home/yc/app/ThreadMentor/include
6 TMLIB = /home/yc/app/ThreadMentor/Visual/libthreadclass.a
7 TMLIB_NV = /home/yc/app/ThreadMentor/NoVisual/libthreadclass.a
8
9 OBJ_FILE = thread.o boat-monitor.o thread-main.o
10 EXE_FILE = prog5
11
12 ${EXE_FILE}: ${OBJ_FILE}
13 | ${CC} ${FLAGS} -o ${EXE_FILE} ${OBJ_FILE} ${TMLIB} -lpthread
14
15 boat-monitor.o: boat-monitor.cpp
16 | ${CC} ${DFLAGS} ${IFLAGS} ${CFLAGS} -c boat-monitor.cpp
17
18 thread.o: thread.cpp
19 | ${CC} ${DFLAGS} ${IFLAGS} ${CFLAGS} -c thread.cpp
20
21 thread-main.o: thread-main.cpp
22 | ${CC} ${DFLAGS} ${IFLAGS} ${CFLAGS} -c thread-main.cpp
23
24 noVisual: ${OBJ_FILE}
25 | ${CC} ${FLAGS} -o ${EXE_FILE} ${OBJ_FILE} ${TMLIB_NV} -lpthread
26
27 clean:
28 | rm -f ${OBJ_FILE} ${EXE_FILE}
29
```

b) Make and test your code with ThreadMentor visualization.

Activities ThreadMentor: Thread Visualization System Jan 17 13:36

Thread Status

	Running	Joined	Blocked	Suspended	Terminated
Main		☹️			
Boat			👤👤👤		
Cannibal1			👤		
Cannibal2			👤		
Cannibal3			👤		
Missionary1	😊				
Missionary2	😊				

```

Missionary 3 starts
Missionary 4 starts
Missionary 5 starts
Cannibal 1 arrives
Cannibal 2 arrives
Missionary 1 arrives
Cannibal 3 arrives
Missionary 2 arrives
Missionary 4 arrives
Missionary 3 arrives
Missionary 5 arrives
MONITOR(1): three cannibals (3, 1, 2) are select
***** Boat load (1): Passenger list ((null))
***** Boat load (1): Completed
Cannibal 1 arrives
Cannibal 2 arrives
Cannibal 3 arrives
***** The boat is ready
  
```

ThreadMentor: Thread Visualization System

File Options Help

ThreadMentor: Thread Visualization System v1.3

Window Name
Thread Status

- Displays
- Mutex
- Semaphore
- Monitor
- RWLock
- Barrier
- Channel

Non-List Views

- Thread Status
- History Graph
- Thread Hierarchy

Communications

Paused 2.0

second interval

Step Resume Pause