

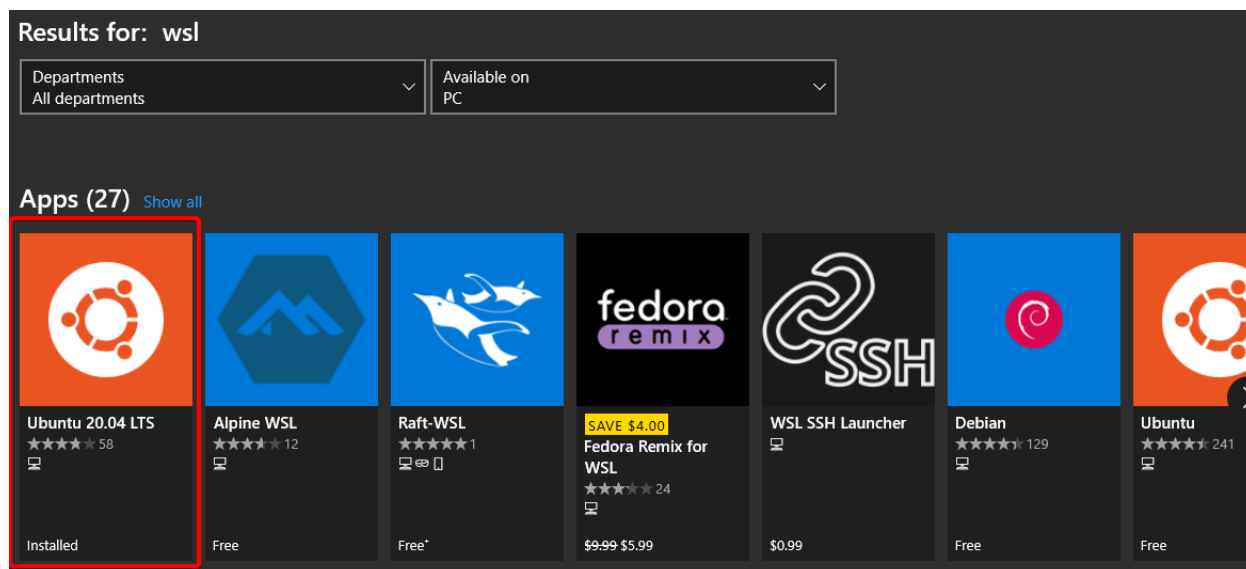
Simulator Setup

Table of Contents

1.	Installing Linux for Windows 10 (WSL)	2
2.	Download project files from GitHub	3
3.	Installing libraries for WSL	5
4.	Installing VcXsrv Windows X Server	6
5.	Configuring XLaunch	8
6.	Compiling the code with makefile and running the simulator	11

1. Installing Linux for Windows 10 (WSL)¹

- Follow steps 1 to 6 in this article: <https://docs.microsoft.com/en-us/windows/wsl/install-win10> (These steps may require multiple PC reboots)
- For step 6, install the latest version of Ubuntu LTS from the Windows store. (currently 20.04)



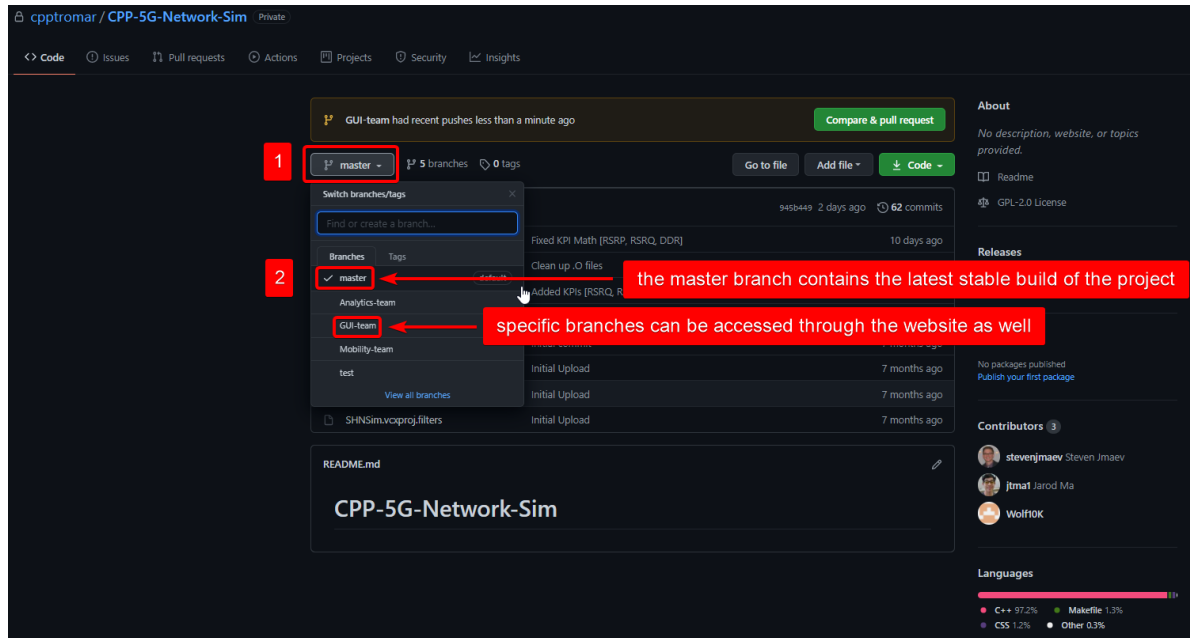
- After successfully installing Ubuntu 20.04 LTS, open the program through the app store. It will prompt the user to create a password. This password will be used in a future step so be sure to write it down somewhere.

¹ Linux machine or Mac, this step can be skipped.

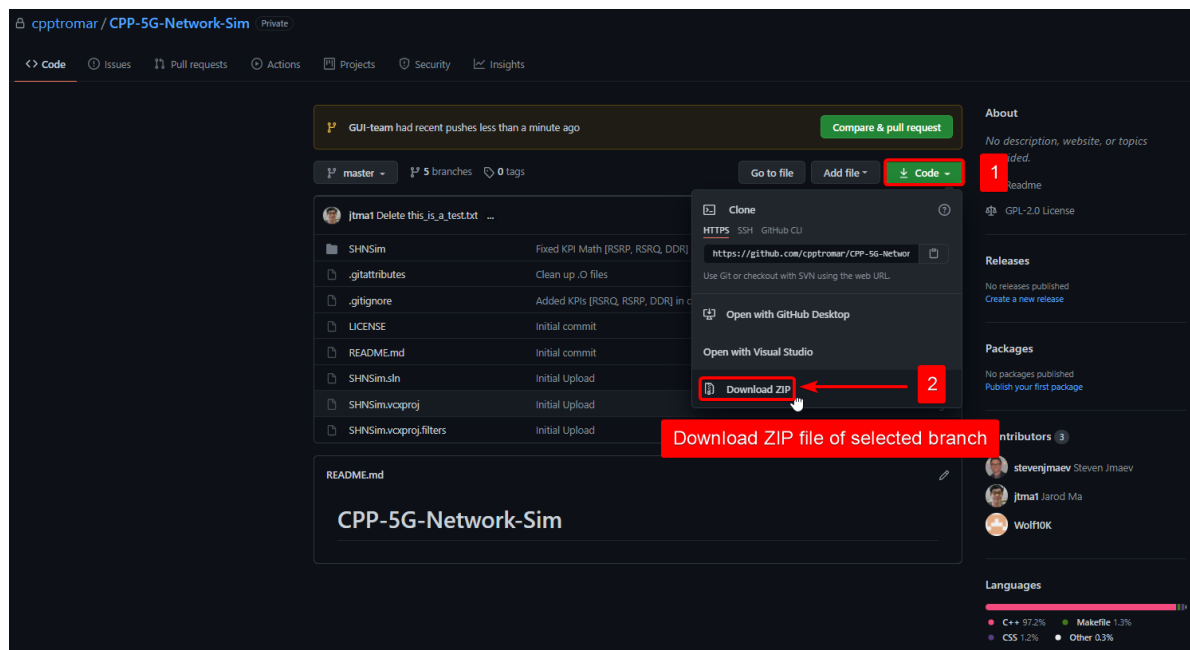
2. Download project files from GitHub²

<https://github.com/cpptromar/CPP-5G-Network-Sim>³

a. Choose version to download



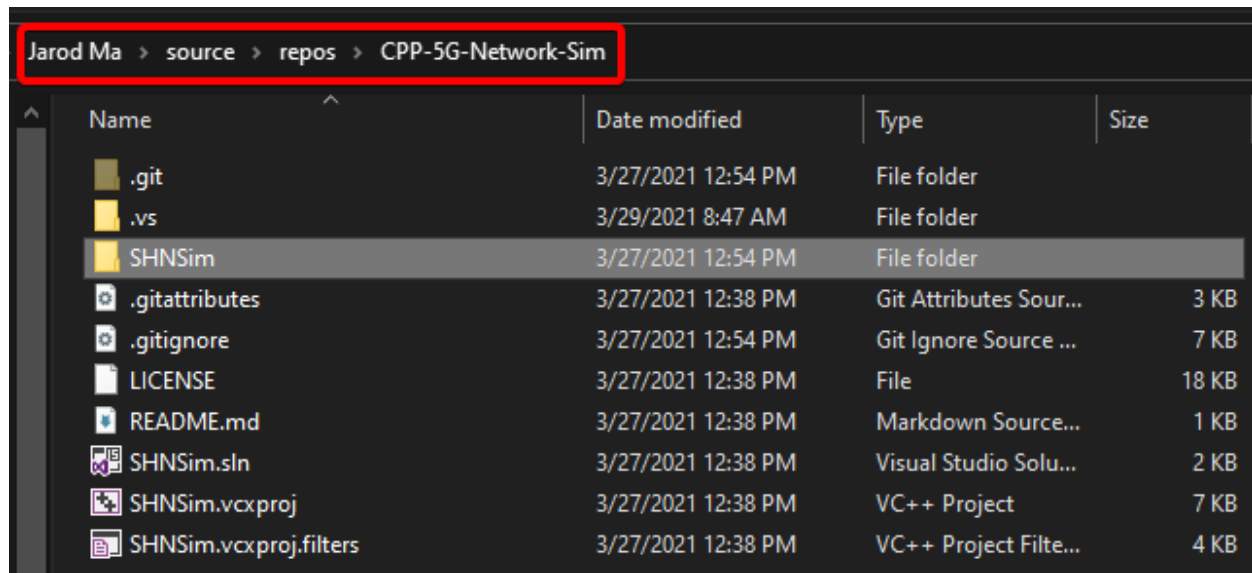
b. Download ZIP file



² A basic tutorial related to how GitHub works can be found here: <https://youtu.be/w3jLUJ7DT5E>

³ This link requires permission to access the files and may change in the future depending on the owner of the repo.

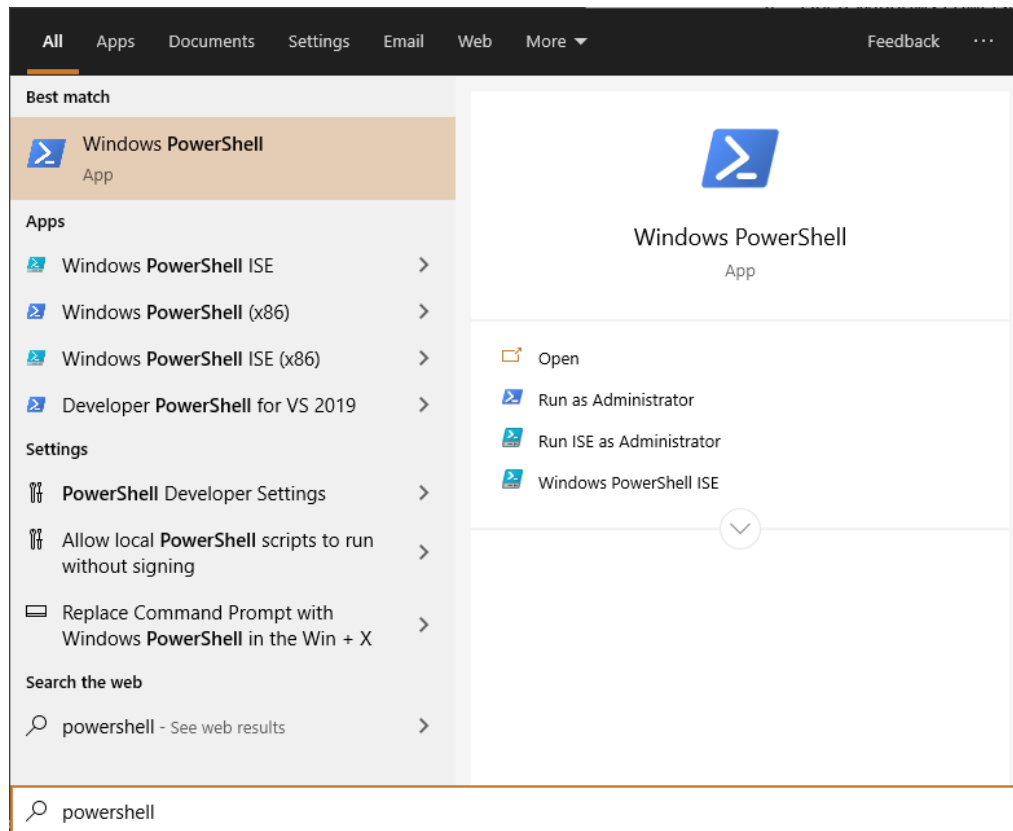
c. Extract ZIP file to a folder⁴



⁴ We will be running all the Linux commands within the SHNSim folder. Be sure to extract the ZIP files to a folder instead of attempting to run the commands within the ZIP.

3. Installing libraries for WSL

a. Open Windows PowerShell⁵



b. Change directory to wherever the SHNSim folder is located

```
cd [path\to\SHNSim_folder]
```

c. Run WSL

```
wsl
```

d. Install all required packages by running the following commands

GTK & Dependency Libraries (for GUI)

```
sudo apt-get update
sudo apt-get install libgtk-3-dev
sudo apt-get install libgtk-3-0
sudo apt-get install pkg-config
sudo apt-get install libcairo2-dev
```

Makefile & G++ Libraries (for compiling)

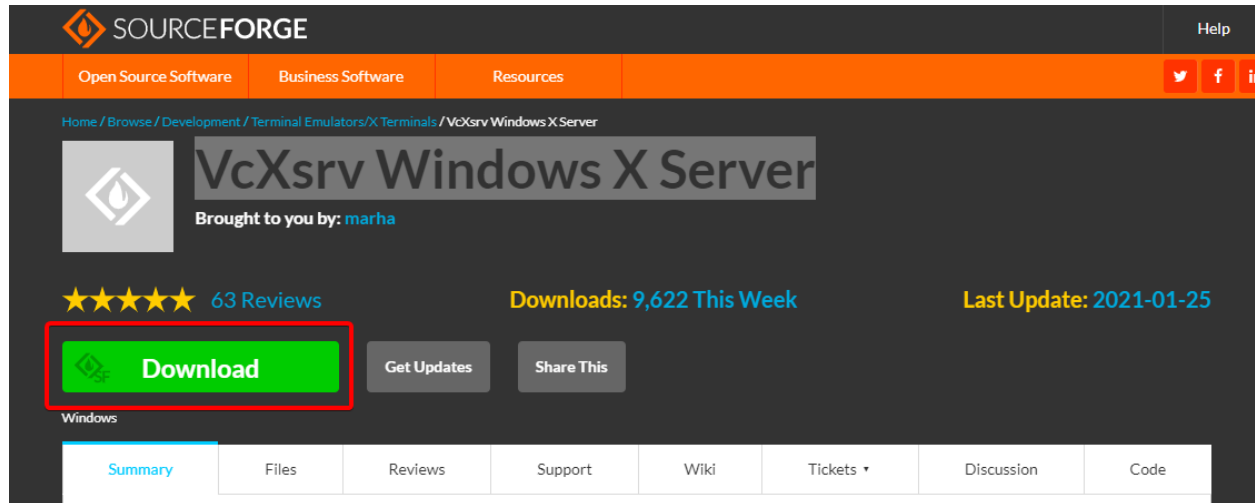
```
sudo apt install make
sudo apt install make-guile
sudo apt install g++
```

⁵ For Mac/Linux/Ubuntu, use the terminal application to run command lines. The “wsl” command does not need to be executed because it is used to start Windows Subsystem for Linux (WSL).

4. Installing VcXsrv Windows X Server⁶

- a. Download VcXsrv from their official website and run their installer (may require a PC reboot)

<https://sourceforge.net/projects/vcxsrv/>



- b. Run this command in PowerShell

```
sudo nano ~/.bashrc
```

- c. Enter your password that you set earlier when running WSL for the first time.

⁶ This application is used as a virtual display when running the “./output” command. Mac/Linux/Ubuntu users may require a different program to support displaying the simulator. Refer to the C++ GTK documentation for more information (<https://www.gtk.org/docs/language-bindings/cpp>).

- d. Use the down arrow key to navigate to the bottom of the bash file and add the following code:

```
export DISPLAY="`grep nameserver /etc/resolv.conf|sed 's/nameserver //'':0'"
```

```
GNU nano 4.8 /home/j/.bashrc
test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
alias ls='ls --color=auto'
#alias dir='dir --color=auto'
#alias vdir='vdir --color=auto'

alias grep='grep --color=auto'
alias fgrep='fgrep --color=auto'
alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "${[ $? = 0 ] && echo terminal || echo error}" "${history|tail -n1|sed -e '\''>

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ] then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

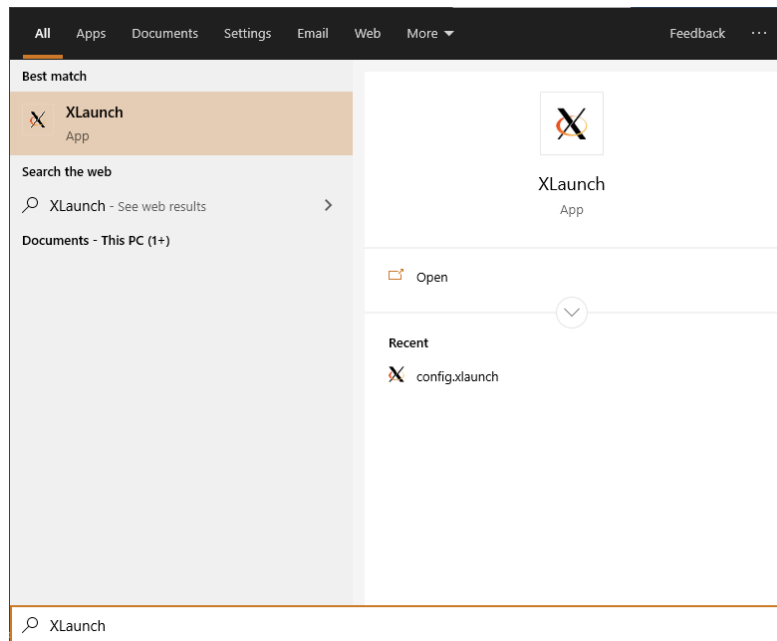
# Launch thing
export DISPLAY="`grep nameserver /etc/resolv.conf|sed 's/nameserver //'':0'"

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      M-U Undo        M-A Mark Text
^X Exit          ^R Read File    ^N Replace      ^U Paste Text   ^T To Spell     ^_ Go To Line    M-E Redo        M-G Copy Text
```

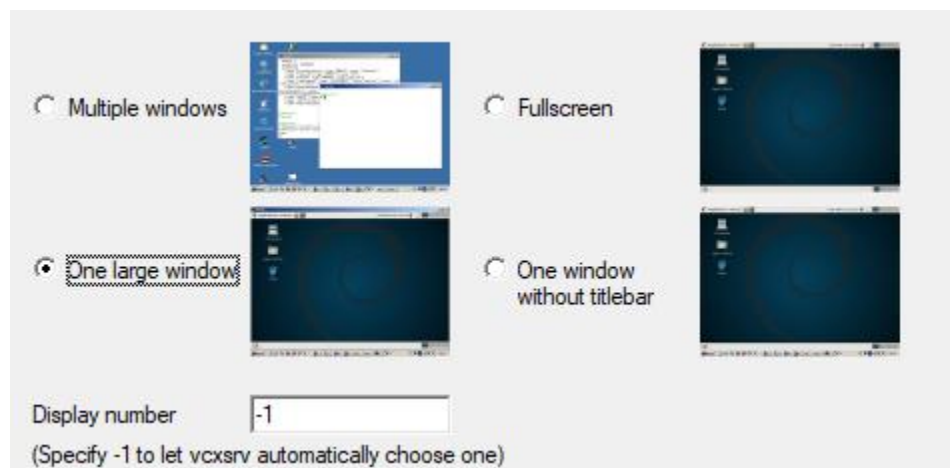
- e. Save and close the bash file using these three key combinations in order: CTRL+O, Enter, CTRL+X
 f. Close and re-open PowerShell

5. Configuring XLaunch⁷

- a. Open XLaunch through windows explorer

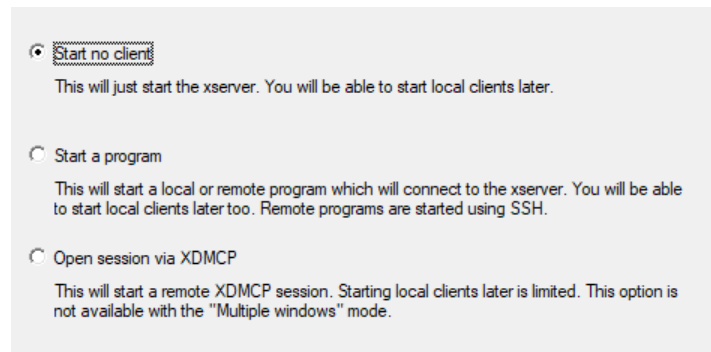


- b. Select "One Large Window" and then press "Next"



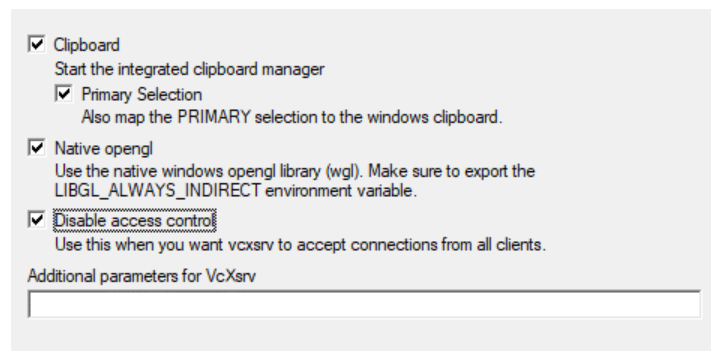
⁷ The most efficient way to use XLaunch is to create a .config file that will be opened every time the simulator is opened. These following steps will show the process of generating a .config file using the XLaunch setup wizard.

- c. Select "Start no client" and then press "Next"



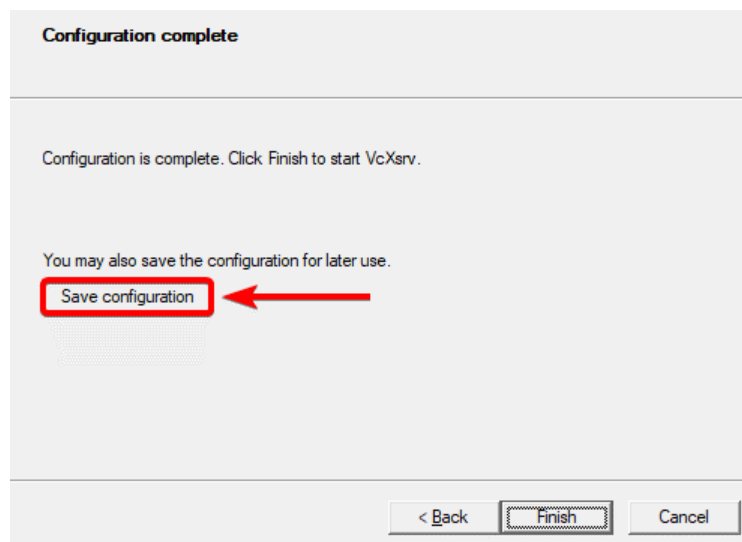
The screenshot shows a configuration window with three radio button options. The first option, "Start no client", is selected and highlighted with a red rectangle. Below it, a description states: "This will just start the xserver. You will be able to start local clients later." The second option is "Start a program", with a description: "This will start a local or remote program which will connect to the xserver. You will be able to start local clients later too. Remote programs are started using SSH." The third option is "Open session via XDMCP", with a description: "This will start a remote XDMCP session. Starting local clients later is limited. This option is not available with the 'Multiple windows' mode."

- d. Check "Disable Access Control" and then press "Next"

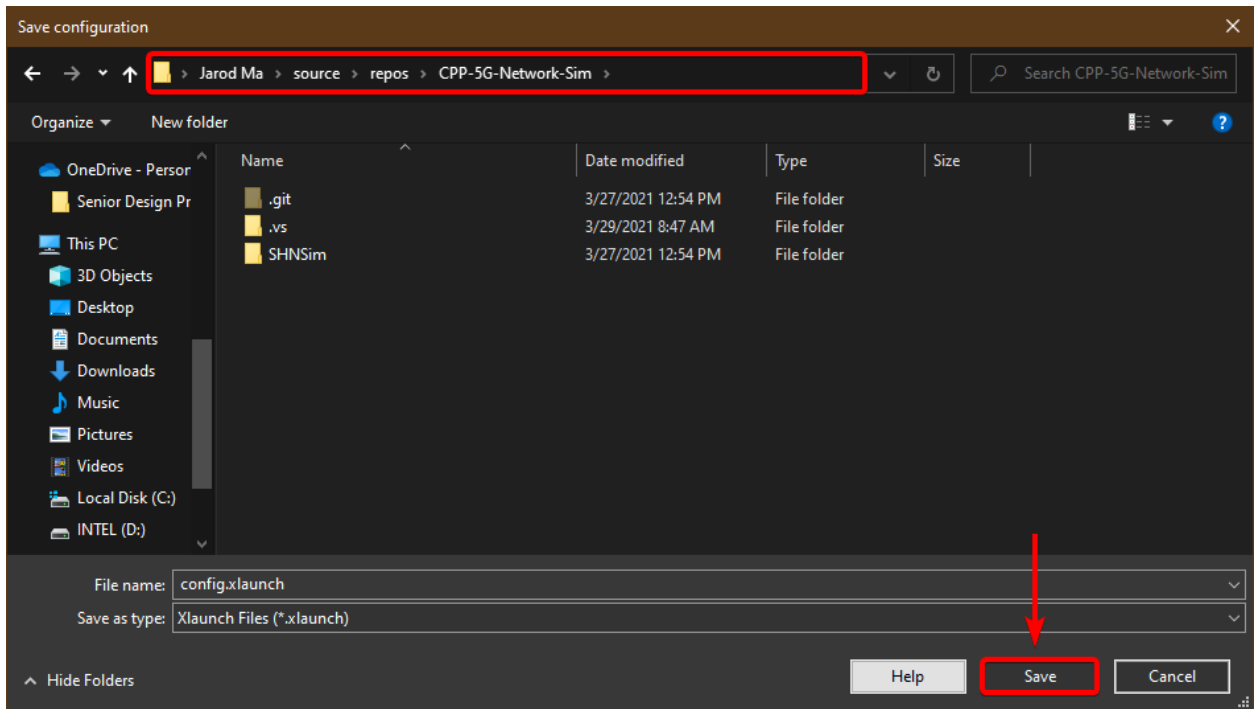


The screenshot shows a configuration window with four checked checkboxes. The first is "Clipboard" with the sub-option "Primary Selection" also checked. The second is "Native opengl". The third is "Disable access control", which is highlighted with a red rectangle. Below the checkboxes, there is a text field labeled "Additional parameters for VcXsrv".

- e. Save configuration (.config file) to somewhere accessible. (e.g. Desktop or Project location)



The screenshot shows a "Configuration complete" dialog box. It contains the text: "Configuration is complete. Click Finish to start VcXsrv." and "You may also save the configuration for later use." Below this text, there is a button labeled "Save configuration", which is highlighted with a red rectangle and a red arrow pointing to it. At the bottom of the dialog, there are three buttons: "< Back", "Finish", and "Cancel".



6. Compiling the code with makefile and running the simulator⁸

- a. Open XLaunch through the configuration file. This will open a blank window.
- b. Change directory to wherever the SHNSim folder is located

```
cd [path\to\SHNSim_folder]
```

- c. Run WSL

```
wsl
```

- d. Compile the code with makefile

```
make output  
make clean
```

- e. Output to Display

```
./output
```

⁸ These steps should be executed every time the simulator's code is modified. To run the simulation without compiling with makefile: Opening Xserver, changing the directory, running WSL, and then using “./output” is sufficient.