g++ with Linux

Every program you submit as source files must run successfully on the SEASnet Linux server cs32.seas.ucla.edu using the g32 command we have set up, which is merely a special way of running g++ that catches more kinds of errors during execution.

What we're presenting here is a minimal set of instructions for helping a Windows or Mac user transfer a C++ program to that SEASnet Linux server to build and run. There are other ways to do this than what we show; the more you know about Linux, the more you can simplify or eliminate certain steps.

Let's assume you have a Visual C++ or an Xcode project consisting of the three source files from the Visual C++ 2019 writeup or the Xcode on a Mac writeup: VoiceMail.h, VoiceMail.cpp, and testVoiceMail.cpp. (Note: It is inconvenient in Linux if the name of a file contains space characters, so avoid giving your files such names.) Here's what to do to run the program on the SEASnet cs32 Linux server:

1. First, note that you will be able to log into that SEASnet Linux server only from
    a. a machine on the UCLA campus network, or
    b. the Windows-based remote SEASnet Terminal Server; or
    c. another machine (e.g., your own computer off-campus), provided you first connect to the campus VPN (Virtual Private Network) server, which makes it look like your machine is on the campus network.
2. Create a zip file containing your three source files. Let's suppose you named the zip file VM.zip.
3.
    a. (Windows users) Put the zip file on your Windows desktop on a SEASnet machine. Make sure you copy the file; don't just create a shortcut linking to it. Note: A few people each quarter find that step 7 below produces produces the error message unzip: cannot find or open Desktop/VM.zip, Desktop/VM.zip.zip or Desktop/VM.zip.ZIP. If that happens to you, return to this step and copy your zip file to Z:\Desktop or to \\labsamba2.seas.ucla.edu\yourUserName\Desktop instead of just dragging the file's icon to the desktop.
    b. (Mac users) Put the zip file onto your desktop. Launch the Terminal application. In Terminal, run the command scp Desktop/VM.zip yourSEASaccount@lnxsrv06.seas.ucla.edu:Desktop and enter your SEASnet account password when prompted. This copies your file from your Mac to the SEASnet cs32 Linux server.
    c. If that command results in a scary warning about DNS spoofing, then run the command ssh yourSEASaccount@lnxsrv06.seas.ucla.edu date and answer yes to the question it asks. Then try the scp command again.
4.
    a. (Windows users) From your Windows desktop on a SEASnet machine, double click the icon labelled "Putty" (not "SSH LnxSrv"). Type cs32.seas.ucla.edu for the Host Name in the dialog box. If a dialog box

titled "PuTTY Security Alert" appears, click its Yes button. In the PuTTY console window (titled something like "lnxsrv06.seas.ucla.edu - PuTTY"), log in to the Linux server with your SEASnet login name. When prompted for your password, type it in; the characters you type for your password will not produce anything on the screen, not even bullets. If there is no Putty icon, then double click on either "SSH LnxSrv" or "SSH UGrad" or "SSH Grad". Log in. After you've typed your password, then in response to the command prompt, type ssh yourSEASaccount@lnxsrv06.seas.ucla.edu and your password again when prompted.

b. (Mac users) From the Terminal application, run the command ssh yourSEASaccount@lnxsrv06.seas.ucla.edu and enter your SEASnet account password when prompted. This logs you into the Linux server.

5. You will now be interacting with the command interpreter program (the shell), whose default prompt is something like bash-4.4$ or [yourname@lnxsrv06 ~]$. For each command you type, the shell will execute it and then prompt you for the next command. The shell is case-sensitive, so pay attention to the distinction between lower and upper case. About two or three people per quarter discover that a mistake was made when their SEASnet account was set up. To verify you're not one of them, run the following command to list the contents of your desktop:

   ```
   ls Desktop
   ```

   If it produces the output ls: cannot access Desktop: No such file or directory, or if it does not show the name of the zip file you put on your desktop (e.g., VM.zip), then either you did not follow these instructions correctly or you are one of the rare people whose SEASnet account setup wasn't done quite right. In the latter case, contact the SEASnet Help Desk in Boelter 2684 or at help@seas.ucla.edu; don't do so in the former case, or you'll be wasting their time with your foolishness.
6. Our special way of running g++ is via a new command, g32, which you will have to set up. If you have never previously run the following command, run it now; you will never have to run it again. If you do run it again, it will confirm that you are set up for running g32 on the SEASnet Linux server.

   ```
   curl -s -L
   https://drive.google.com/file/d/1wAYNlKqXXHRp1Kuf9eby2TmDcePilHq5/view?usp=sharing | bash
   ```

   NOTE: Some students have problem with this step for various reasons. If you do, try these steps instead:

   1. Copy the contents of the Google drive file.

2. Within the Linux server, use the command pico to open up a text editor with a new file name (for example, water) so it looks like this:

pico water

3. Paste with a right click into the pico text editor.

4. Hit Ctrl-X to exit the text editor. Type Y, then Enter, to save your changes.

5. Re-run the curl command like this:

curl -s -L water | bash

6. If you're in lnxsrv07, then type g32 on its own and it should say you're missing a main() function, which means it works. =]

7. Unbundle the zip file into its own directory called, say, proj0:

   ```
   unzip -j -a -d proj0 Desktop/VM.zip
   ```

8. Make proj0 the current directory (i.e., the default directory for now in which files will be found or created):

   cd proj0

   Like many Linux commands, if this command works, it doesn't say anything, so the shell would then just print its next prompt.

9. Verify that the expected three files are present by listing the contents of the current directory:

   ```
   ls
   ```

   File names in Linux are case-sensitive. You'll have a problem if one of your .cpp files says #include "VoiceMail.h" but the .h file is named voicemail.h. You could use the move command to change the name of the file:

   ```
   mv voicemail.h VoiceMail.h
   ```

10. Build an executable file from the source files. If we would like the executable file to be named vm, we'd say

    ```
    g32 -o vm *.cpp
    ```

The *.cpp saves us typing individual file names by matching all the files whose names end in .cpp. Notice that we do not list any .h files. (You don't have to know this, but the setup process in step 6 modified the file ~/.profile to create a bash function that causes the above command to be executed as if it were `/usr/local/cs/bin/g++ -std=c++14 -Wall -Wextra -Wno-sign-compare -Werror=return-type -Wl,--rpath=/usr/local/cs/lib64 -fsanitize=address -fsanitize=undefined -fsanitize=bounds -fno-omit-frame-pointer -o vm *.cpp` instead. The `/usr/local/cs/bin/g++` invokes g++ version 6.3.0; the SEASnet server's default is version 4.8.5. The `-std=c++14` enables C++14 language features. The `-Wall` option asks the compiler to warn you about many questionable constructs; the `-Wextra -Wno-sign-compare` asks for warnings about even more questionable constructs. The `-Werror=return-type` causes a certain warning to be treated as an error. The other added options cause certain runtime errors to terminate your program with an error message instead of silently continuing and wreaking havoc.)

Compiler diagnostic messages are of the form

```
fileName:lineNumber:columnNumber:message
```

If the compiler detects any problems that you want to fix, then since we're assuming you're doing your primary development using Visual C++ or Xcode, you should make changes to your original Windows or Mac files. After checking in Visual C++ or Xcode that the modified program works as you expect it to, go back to step 1.

11. To execute the program vm that you built, you'd just say

    ```
    vm
    ```

    (If that doesn't work, try

    ```
    ./vm
    ```

    instead.)

12. To exit the shell, say

    ```
    exit
    ```

If you want to examine a file under Linux, a simple text editor you can use is Nano.

```
nano testVoiceMail.cpp
```

You can navigate with the arrow keys. The bottom two lines of the display show you some commands you can type. For example, control-C (indicated in the bottom display as ^C) shows you what line number the cursor is on. Control-O saves any changes you make to the file, and control-X exits the editor.

We strongly recommend that you don't test your program using g32 just once, only after you're satisfied with it under Visual C++ or Xcode. Instead, do it periodically during your development, and certainly do it when you're trying to find an elusive bug. Sometimes people have spent hours trying to track down a problem during execution using Visual C++ when the g32 command would have immediately given them a warning that pointed to the mistake. Also, sometimes a program that appeared to work correctly under Visual C++ or Xcode relied on undefined behavior, but happened to get lucky for the student in a way that wasn't reproduced when we tested the program; testing using g32 might have shown different behavior, which would have suggested to the student that something was amiss.