COMP3500 – Frequently Asked Questions
Project 2 – An Introduction to OS/161

1. Some of the files aren't in the files section, and we can't see the discussion you showed in class today. The files are:

   cross compiler: `cs161-gcc-1.5.tar`
   special gdb: `cs161-gdb-1.5.tar`

   How do we install them after we install the os and tool chain?

   Answer: The above two files are available in the Canvas system. The OS161 installation packages can be found here: `Canvas->Files->Projects->OS161`

   The instruction on how to install the above two packages can be found in "The MIPS toolchain for os161.txt". Run the `tar` command to unzip the tarred and compressed file. Then, change to the directory where you just unzipped the files. Last, run `./toolbuild.sh`

2. After I download `cs161-binutils-1.5`, I ran the `` `./toolbuild.sh` `` and I received an error at the end of the command. How to solve this problem? (Contributed by Collin Pike, Fall'18)

   Answer: You must first setup CFLAGS as follows and then run "`./toolbuild.sh`"
   `%export CFLAGS="-g -O2 -Wno-error"`

   After you build binutils, you will receive the following message (i.e., Done).

   ```
     done; \
   fi
   make[3]: Entering directory `/home/cse_h1/xzq0001/mycs161/cs161-binutils-1.5/libiber
   ty/testsuite'
   make[3]: Nothing to be done for `install'.
   make[3]: Leaving directory `/home/cse_h1/xzq0001/mycs161/cs161-binutils-1.5/libibert
   y/testsuite'
   make[2]: Leaving directory `/home/cse_h1/xzq0001/mycs161/cs161-binutils-1.5/libibert
   y'
   make[1]: Nothing to be done for `install-target'.
   make[1]: Leaving directory `/home/cse_h1/xzq0001/mycs161/cs161-binutils-1.5'
   ******** Creating symlinks ********
   addr2line ar as c++filt ld nm objcopy objdump ranlib readelf size strings strip
   ******** Done ********
   xzq0001@tux058:~/mycs161/cs161-binutils-1.5$ cd ..
   ```

3. While trying to configure the cs161-gdb for Project 2 we are encountering a build error: "no termcap library found". We built the various libraries in the correct order and even wiped the libraries and rebuilt them to no success. Is there any advice you could give us for solving this error? I just tried to attach the GDB to my kernel and I am having issues with what looks to be that the GDB is not attaching to the kernel:

My Kernel Window:

```
cdp0037@tux054:~$ cd ~/cs161/root/
cdp0037@tux054:~/cs161/root$ ./sys161 -w kernel
sys161: System/161 release 1.14, compiled Sep  6 2018 21:23:58
sys161: Waiting for debugger connection...
```

My GDB windows looks like:

```
bash-4.2$ cd ~/cs161/root/
bash-4.2$ cs161-gdb kernel
GNU gdb 6.6
Copyright (C) 2006 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "--host=x86_64-unknown-linux-gnu --target=mips-elf"...
(gdb) target remote unix:.sockets/gdb
unix:.sockets/gdb: Connection refused.
```

How to address this issue? (Contributed by Collin Pike, Fall'18)

Answer: Both your run window and debug window must be running on the same Tux machine. More specifically, when you log into a tux machine for a second time for your debug window, please choose the same Tux machine where your running window is residing. For example, when you first log into tux052 for your run window, you must log into tux052 for the second time for your debug window.

4. While trying to configure the cs161-gdb for Project 2 we are encountering a build error: "no termcap library found". We built the various libraries in the correct order and even wiped the libraries and rebuilt them to no success. Is there any advice you could give us for solving this error?

    Answer: This problem can be solved by installing ncurses-dev. The distro came installed with ncurses-lib, which was not enough. Please use the following command to install the package:

    ```
    yum install ncurses-devel
    ```

5. When I try to run make depend to make the kernel, I get the following error:

    ```
    /bin/sh: cs161-gcc: command not found
    make: *** [depend] Error 127
    ```

    What should I do to fix this?

Answer: Use 'echo $PATH' to check your $PATH. Your $PATH must contains `~/cs161/bin` You can run the following command to setup your path.

```
export PATH=~/cs161/bin:$PATH
```

You may need to add that statement to your `.bashrc file`.
The specific file is `~/.bashrc`
You can add "`export PATH=~/cs161/bin:$PATH`" to the last line.

6. When I tried to run cs161-gdb to debug my os161 kernel, I encountered the following error message:

```
warning: unrecognized item "timeout" in "qSupported"
response.
```

How to fix this problem?

Answer: The sys161 you installed is incompatible with your cs161-gdb. Please download the correct versions for the tool chain as follows:

1)   OS/161: `os161-1.10.tar.gz`
2)   tool chain: `cs161-binutils-1.4.tar`
3)   cross compiler: `cs161-gcc-1.5.tar`
https://dl.dropboxusercontent.com/u/24238235/cs161-gcc-1.5.tgz

4)   special gdb: `cs161-gdb-1.5.tar`
https://dl.dropboxusercontent.com/u/24238235/cs161-gdb-1.5.tgz

5)   sys161 MIPS emulator: `sys161-1.14.tar`
https://dl.dropboxusercontent.com/u/24238235/sys161-1.14.tar.gz

Please build and install sys161 version 1.14 instead of version 1.12.

7. I keep getting "command not found" error when I try "sys161 - kernel" in the cs161/root. I get this same error when I try to run "cs161-gdb kernel" in the run window. I noticed that ~cs161/bin has cs161-gcc but it doesn't contain cs161-gdb. I don't know if that should be there or not.

Answer: If executable file `sys161` isn't found in `~/cs161/root`, then you haven't yet successfully compiled and built the cs161-gdb debugger. You must build cs161-gdb first before running cs161-gdb.

8. When I tried to use the tar command to create, I had the following error message:
`tar: Cowardly refusing to create an empty archive.`

Answer: Do not forget to put a dot (i.e., .) at the end of the following command.

```
% tar vfcz <group_ID>_asst0.tgz .
```

If you ignore the dot, you will fail in creating a tarball.

9. In step 10.8, we are asked to add debugging messages. How can we add debug messages?

   Answer: You print debug messages to the console using the DEBUG() macro. A sample source code is given below:

   ```
   DEBUG(DB_VM, "VM free pages: %u\n", free_pages);
   ```

   In the above code, `DB_VM` is a flag defined in `src/kern/include/lib.h`
   If the debug DB_VM flag is set, the debug message will be printed on the console.

10. How do we know what to put in the parameters of DEBUG() ? Also, do we put the comments in main.c ?

    Answer: You can put the DEBUG() statements in `main.c`. You may add any debugging message to demonstrate that you understand how make use of DEBUG() to debug your OS/161.

11. We make 10 different variations of

    ```
    #if 0
    #define DEBUG(d, fmt, ...) ((dbflags & (d)) ? kprintf(fmt,
    __VA_ARGS__) : 0)
    #else
    #define DEBUG(d, fmt, args...) ((dbflags & (d)) ? kprintf(fmt, ##args) :
    0)
    #endif
    ```

    We understand that we can put the almost anywhere. We just don't understand the context of the message itself. Is it supposed to be variations of the sample?

    Answer: The macro "DEBUG()" allows you to add debugging related messages in your OS/161 source code. Note that "DEBUG()" is much better than "kprintf()" when it comes to printing debugging messages, because DEBUG() allows you to easily configure what debugging messages should be enabled and what debugging messages should be disabled.

12. I realize this is bad practice but for the sake of the project we're not able to commit our current kernel changes to cvs as root. Is there a setting or an option we can enable to allow us to commit as root just to finish this project?

    Error message: `cvs [commit aborted]: 'root' is not allowed to commit files`

Answer: When committing a permanent change, CVS makes a log entry of who committed the change. If you are committing the change logged
 in as "root" (not under "su" or other root-priv giving program), CVS cannot determine who is actually making the change. As such, by default, CVS disallows changes to be committed by users logged in as "root".
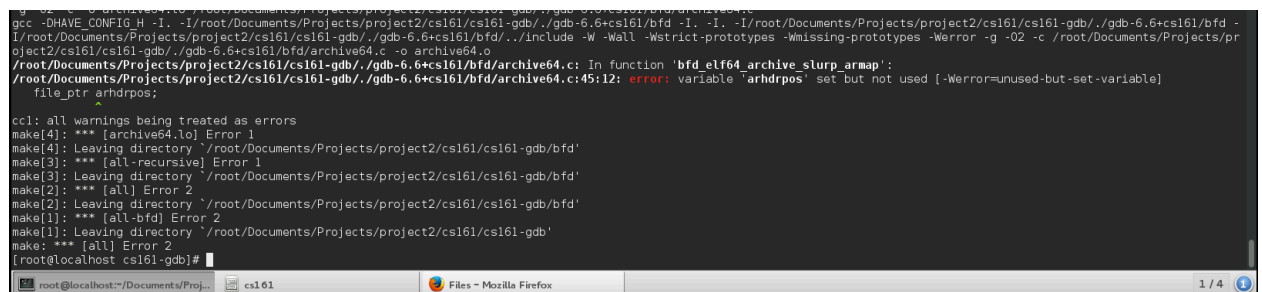
There are two solutions to this problem.

- **Solution 1:** It is worth noting that I recommend against this solution. You can disable this option by passing the --enable-rootcommit option to 'configure' and recompiling CVS. On some systems this means editing the appropriate 'config.h' file before building CVS (See also the reference below).

- **Solution 2:** Please note that this solution is better than the first one. When you need to install any package, you use the superuser (i.e., root) privilege. In all the other cases, you use your normal user privilege to let CVS know who is committing changes. If you are sharing one Linux machine with the other two group members, only one can be login as root. Each member should have his or her individual working directory. Developing OS161 on the shared machine using a normal user privilege is a practical way to collaborate with your group members.

  Reference: https://www.linuxquestions.org/questions/linux-newbie-8/cvs-%5Bcommit-aborted%5D-root-is-not-allowed-to-commit-files-891643-print/

13. While running ./toolbuild.sh for the special gdb I get the following error:

```
"error: variable 'arhdrpos' set but not used [-
Werror=unused-but-set-variable] file_ptr arhdrpos;"
```


There were several other variables with similar problems. I think the issue is that the warnings are being treated as errors. How to fix this compilation problem?

Answer: This problem can be fixed by setting up the compilation flag variable `CFLAGS`. Please follow the three steps below:

Step 1: extract the special gdb tarball
Step 2: `%export CFLAGS="-g -O2 -Wno-error"`

Step 3. run the `toolbuild.sh` script

14. We changed CFLAGS as described in Question 12, but we got an error that we've changed CFLAGS and it quits after that.

    Answer: After you change CFLAGS, you must delete all of the `config.cache` files and rebuild everything. To delete the `config.cache` files, please carry out the following steps:

    Step 1: `cd ~/cs161`
    Step 2: `rm -f ./*/config.cache`
    Step 3: `rm -f ./*/*/config.cache`

    Once you finish the above steps, you should rebuild everything in order.