Lab03-Report

Course : ICSI 424

Name : Seoyeon Choi

gitHub : Seoyeon-bot

Before we change anything about myprintenv.c

```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ls
README.md               cap_leak.dSYM           lab03-env-setuid.pdf
cap_leak                catall.c                myenv.c
cap_leak.c              lab03-env-setuid-2.pdf  myprintenv.c
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ls-l
zsh: command not found: ls-l
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ls -l
total 3344
-rw-r--r--@ 1 choeseoyeon  staff     1532 Sep 12 05:39 README.md
-rwxr-xr-x  1 choeseoyeon  staff    33648 Sep 12 12:30 cap_leak
-rw-r--r--@ 1 choeseoyeon  staff      761 Sep 12 15:13 cap_leak.c
drwxr-xr-x@ 3 choeseoyeon  staff       96 Sep 12 12:30 cap_leak.dSYM
-rw-r--r--@ 1 choeseoyeon  staff      471 Sep 12 05:39 catall.c
-rw-r--r--@ 1 choeseoyeon  staff  1223364 Sep 12 16:17 lab03-env-setuid-2.pdf
-rw-r--r--@ 1 choeseoyeon  staff   429653 Sep 12 15:20 lab03-env-setuid.pdf
-rw-r--r--@ 1 choeseoyeon  staff      180 Sep 12 05:39 myenv.c
-rw-r--r--@ 1 choeseoyeon  staff      418 Sep 12 16:18 myprintenv.c
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % cat myprintenv.c
```

2.1 Task 1: Manipulating Environment Variables

```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % gcc myprintenv.c -o myprintenv
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % gcc myprintenv.c -o myprintenv
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ls
README.md               catall.c                myprintenv
cap_leak                lab03-env-setuid-2.pdf  myprintenv.c
cap_leak.c              lab03-env-setuid.pdf
cap_leak.dSYM           myenv.c
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ./   myprintenv

zsh: permission denied: ./
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ./   myprintenv

zsh: permission denied: ./
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % .\   myprintenv

zsh: command not found: .\tmyprintenv
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ./myprintenv
__CFBundleIdentifier=com.apple.Terminal
TMPDIR=/var/folders/37/rntj73sd0y96f9441_38454w0000gn/T/
XPC_FLAGS=0x0
LaunchInstanceID=22431546-C138-4E73-A4A6-1EBC5D481BA2
TERM=xterm-256color
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.2w8xHHO1Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=0
TERM_PROGRAM=Apple_Terminal
TERM_PROGRAM_VERSION=447
TERM_SESSION_ID=799C7D92-7ECB-4585-BADE-CBBA72606CD9
SHELL=/bin/zsh
HOME=/Users/choeseoyeon
LOGNAME=choeseoyeon
USER=choeseoyeon
PATH=/Users/choeseoyeon/anaconda3/bin:/Users/choeseoyeon/anaconda3/condabin:/usr/local/bin:/usr/lo
cal/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framewo
rk/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/us
r/local/share/dotnet:~/.dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptexd/c
odex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/us
r/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/Users/cho
eseoyeon/.zsh/misc
SHLVL=1
PWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main
OLDPWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3
HOMEBREW_PREFIX=/usr/local
HOMEBREW_CELLAR=/usr/local/Cellar
HOMEBREW_REPOSITORY=/usr/local/Homebrew
MANPATH=/usr/local/share/man::
INFOPATH=/usr/local/share/info:
CONDA_EXE=/Users/choeseoyeon/anaconda3/bin/conda
_CE_M=
_CE_CONDA=
CONDA_PYTHON_EXE=/Users/choeseoyeon/anaconda3/bin/python
CONDA_SHLVL=1
CONDA_PREFIX=/Users/choeseoyeon/anaconda3
CONDA_DEFAULT_ENV=base
```

// Above shows environmental variable when I run myprintenv.c

```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ./myprintenv printenv PWD
__CFBundleIdentifier=com.apple.Terminal
TMPDIR=/var/folders/37/rntj73sd0y96f9441_38454w0000gn/T/
XPC_FLAGS=0x0
LaunchInstanceID=22431546-C138-4E73-A4A6-1EBC5D481BA2
TERM=xterm-256color
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.2w8xHHO1Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=0
TERM_PROGRAM=Apple_Terminal
TERM_PROGRAM_VERSION=447
TERM_SESSION_ID=799C7D92-7ECB-4585-BADE-CBBA72606CD9
SHELL=/bin/zsh
HOME=/Users/choeseoyeon
LOGNAME=choeseoyeon
USER=choeseoyeon
PATH=/Users/choeseoyeon/anaconda3/bin:/Users/choeseoyeon/anaconda3/condabin:/usr/local/bin:/usr/lo
cal/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framewo
rk/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/us
r/local/share/dotnet:~/.dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptexd/c
odex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/us
r/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/Users/cho
eseoyeon/.zsh/misc
SHLVL=1
PWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main
OLDPWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3
HOMEBREW_PREFIX=/usr/local
HOMEBREW_CELLAR=/usr/local/Cellar
HOMEBREW_REPOSITORY=/usr/local/Homebrew
MANPATH=/usr/local/share/man::
INFOPATH=/usr/local/share/info:
CONDA_EXE=/Users/choeseoyeon/anaconda3/bin/conda
_CE_M=
_CE_CONDA=
CONDA_PYTHON_EXE=/Users/choeseoyeon/anaconda3/bin/python
CONDA_SHLVL=1
CONDA_PREFIX=/Users/choeseoyeon/anaconda3
CONDA_DEFAULT_ENV=base
CONDA_PROMPT_MODIFIER=(base)
LANG=en_US.UTF-8
_=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main/./myprintenv
(base) choeseoyeon@dyn-169-226-220-44 lab03-main %
```

// Above I used printenv PWD to see PWD environment variable (Task 2.1)

```
_=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main/./myprintenv
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ./myprintenv env | grep PWD
PWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main
OLDPWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3
(base) choeseoyeon@dyn-169-226-220-44 lab03-main %
```

// Above is output when I use env | grep PWD to see specific environmental variable(Task 2.1)

```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ./myprintenv export
__CFBundleIdentifier=com.apple.Terminal
TMPDIR=/var/folders/37/rntj73sd0y96f9441_38454w0000gn/T/
XPC_FLAGS=0x0
LaunchInstanceID=22431546-C138-4E73-A4A6-1EBC5D481BA2
TERM=xterm-256color
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.2w8xHHO1Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=0
TERM_PROGRAM=Apple_Terminal
TERM_PROGRAM_VERSION=447
TERM_SESSION_ID=799C7D92-7ECB-4585-BADE-CBBA72606CD9
SHELL=/bin/zsh
HOME=/Users/choeseoyeon
LOGNAME=choeseoyeon
USER=choeseoyeon
PATH=/Users/choeseoyeon/anaconda3/bin:/Users/choeseoyeon/anaconda3/condabin:/usr/local/bin:/usr/lo
cal/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framewo
rk/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/us
r/local/share/dotnet:~/.dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptexd/c
odex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/us
r/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/Users/cho
eseoyeon/.zsh/misc
SHLVL=1
PWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main
OLDPWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3
HOMEBREW_PREFIX=/usr/local
HOMEBREW_CELLAR=/usr/local/Cellar
HOMEBREW_REPOSITORY=/usr/local/Homebrew
MANPATH=/usr/local/share/man::
INFOPATH=/usr/local/share/info:
CONDA_EXE=/Users/choeseoyeon/anaconda3/bin/conda
_CE_M=
_CE_CONDA=
CONDA_PYTHON_EXE=/Users/choeseoyeon/anaconda3/bin/python
CONDA_SHLVL=1
CONDA_PREFIX=/Users/choeseoyeon/anaconda3
CONDA_DEFAULT_ENV=base
CONDA_PROMPT_MODIFIER=(base)
LANG=en_US.UTF-8
_=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main/./myprintenv
(base) choeseoyeon@dyn-169-226-220-44 lab03-main %
```

// Output when I use export.

Exporting an environmental variable means that we set this variable to be visible to other files, so Exporting makes them available to other programs such as child programs or shells.

```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ./myprintenv unset
__CFBundleIdentifier=com.apple.Terminal
TMPDIR=/var/folders/37/rntj73sd0y96f9441_38454w0000gn/T/
XPC_FLAGS=0x0
LaunchInstanceID=22431546-C138-4E73-A4A6-1EBC5D481BA2
TERM=xterm-256color
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.2w8xHHO1Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=0
TERM_PROGRAM=Apple_Terminal
TERM_PROGRAM_VERSION=447
TERM_SESSION_ID=799C7D92-7ECB-4585-BADE-CBBA72606CD9
SHELL=/bin/zsh
HOME=/Users/choeseoyeon
LOGNAME=choeseoyeon
USER=choeseoyeon
PATH=/Users/choeseoyeon/anaconda3/bin:/Users/choeseoyeon/anaconda3/condabin:/usr/local/bin:/usr/lo
cal/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framewo
rk/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/us
r/local/share/dotnet:~/.dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptexd/c
odex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/us
r/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/Users/cho
eseoyeon/.zsh/misc
SHLVL=1
PWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main
OLDPWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3
HOMEBREW_PREFIX=/usr/local
HOMEBREW_CELLAR=/usr/local/Cellar
HOMEBREW_REPOSITORY=/usr/local/Homebrew
MANPATH=/usr/local/share/man::
INFOPATH=/usr/local/share/info:
CONDA_EXE=/Users/choeseoyeon/anaconda3/bin/conda
_CE_M=
_CE_CONDA=
CONDA_PYTHON_EXE=/Users/choeseoyeon/anaconda3/bin/python
CONDA_SHLVL=1
CONDA_PREFIX=/Users/choeseoyeon/anaconda3
CONDA_DEFAULT_ENV=base
CONDA_PROMPT_MODIFIER=(base)
LANG=en_US.UTF-8
_=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main/./myprintenv
(base) choeseoyeon@dyn-169-226-220-44 lab03-main %
```

// Output when I unset.

The unset procedure is used to remove temporary variables from the event, so for example if I set LOGNAME as seoyeon and then if I want to remove that shell variable then I could go back to parent shell and unset LOGNAME as seoyeon. Then the child process shell won't be able to see LOGNAME as seoyeon. And I/O log files when the variables are no longer needed. Variables that are required for the functioning of a Privilege Management for Unix and Linux daemon may not be unset.

2.2 Task 2: Passing Environment Variables from Parent Process to Child Process

```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % gcc myprintenv.c
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ls
README.md               cap_leak.c              lab03-env-setuid-2.pdf  myprintenv
a.out                   cap_leak.dSYM           lab03-env-setuid.pdf    myprintenv.c
cap_leak                catall.c                myenv.c
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ./a.out
__CFBundleIdentifier=com.apple.Terminal
TMPDIR=/var/folders/37/rntj73sd0y96f9441_38454w0000gn/T/
XPC_FLAGS=0x0
LaunchInstanceID=22431546-C138-4E73-A4A6-1EBC5D481BA2
TERM=xterm-256color
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.2w8xHHO1Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=0
TERM_PROGRAM=Apple_Terminal
TERM_PROGRAM_VERSION=447
TERM_SESSION_ID=799C7D92-7ECB-4585-BADE-CBBA72606CD9
SHELL=/bin/zsh
HOME=/Users/choeseoyeon
LOGNAME=choeseoyeon
USER=choeseoyeon
PATH=/Users/choeseoyeon/anaconda3/bin:/Users/choeseoyeon/anaconda3/condabin:/usr/local/bin:/usr/lo
cal/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framewo
rk/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/us
r/local/share/dotnet:~/.dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptexd/c
odex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/us
r/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/Users/cho
eseoyeon/.zsh/misc
SHLVL=1
PWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main
OLDPWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3
HOMEBREW_PREFIX=/usr/local
HOMEBREW_CELLAR=/usr/local/Cellar
HOMEBREW_REPOSITORY=/usr/local/Homebrew
MANPATH=/usr/local/share/man::
INFOPATH=/usr/local/share/info:
CONDA_EXE=/Users/choeseoyeon/anaconda3/bin/conda
_CE_M=
_CE_CONDA=
CONDA_PYTHON_EXE=/Users/choeseoyeon/anaconda3/bin/python
CONDA_SHLVL=1
CONDA_PREFIX=/Users/choeseoyeon/anaconda3
CONDA_DEFAULT_ENV=base
CONDA_PROMPT_MODIFIER=(base)
LANG=en_US.UTF-8
_=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main/./a.out
(base) choeseoyeon@dyn-169-226-220-44 lab03-main %
```

// above I used gcc myprntenv.c to run the file and it automatically created a.out file. I run the a.out file by typing ./a.out.

```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ./a.out > file
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ls
README.md               cap_leak.c              file                    myenv.c
a.out                   cap_leak.dSYM           lab03-env-setuid-2.pdf  myprintenv
cap_leak                catall.c                lab03-env-setuid.pdf    myprintenv.c
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % cat file
__CFBundleIdentifier=com.apple.Terminal
TMPDIR=/var/folders/37/rntj73sd0y96f9441_38454w0000gn/T/
XPC_FLAGS=0x0
LaunchInstanceID=22431546-C138-4E73-A4A6-1EBC5D481BA2
TERM=xterm-256color
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.2w8xHHO1Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=0
TERM_PROGRAM=Apple_Terminal
TERM_PROGRAM_VERSION=447
TERM_SESSION_ID=799C7D92-7ECB-4585-BADE-CBBA72606CD9
SHELL=/bin/zsh
HOME=/Users/choeseoyeon
LOGNAME=choeseoyeon
USER=choeseoyeon
PATH=/Users/choeseoyeon/anaconda3/bin:/Users/choeseoyeon/anaconda3/condabin:/usr/local/bin:/usr/lo
cal/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framewo
rk/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/us
r/local/share/dotnet:~/.dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptexd/c
odex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/us
r/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/Users/cho
eseoyeon/.zsh/misc
SHLVL=1
PWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main
OLDPWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3
HOMEBREW_PREFIX=/usr/local
HOMEBREW_CELLAR=/usr/local/Cellar
HOMEBREW_REPOSITORY=/usr/local/Homebrew
MANPATH=/usr/local/share/man::
INFOPATH=/usr/local/share/info:
CONDA_EXE=/Users/choeseoyeon/anaconda3/bin/conda
_CE_M=
_CE_CONDA=
CONDA_PYTHON_EXE=/Users/choeseoyeon/anaconda3/bin/python
CONDA_SHLVL=1
CONDA_PREFIX=/Users/choeseoyeon/anaconda3
CONDA_DEFAULT_ENV=base
CONDA_PROMPT_MODIFIER=(base)
LANG=en_US.UTF-8
_=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main/./a.out
(base) choeseoyeon@dyn-169-226-220-44 lab03-main %
```

// I saved a.out's output inside the file. So if I type "ls" in the command line to check, I can see that the file is created in lab03-main.

```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ls -l
total 3496
-rw-r--r--@ 1 choeseoyeon  staff     1532 Sep 12 05:39 README.md
-rwxr-xr-x  1 choeseoyeon  staff    33184 Sep 12 16:50 a.out
-rwxr-xr-x  1 choeseoyeon  staff    33648 Sep 12 12:30 cap_leak
-rw-r--r--@ 1 choeseoyeon  staff      761 Sep 12 15:13 cap_leak.c
drwxr-xr-x@ 3 choeseoyeon  staff       96 Sep 12 12:30 cap_leak.dSYM
-rw-r--r--@ 1 choeseoyeon  staff      471 Sep 12 05:39 catall.c
-rw-r--r--  1 choeseoyeon  staff     1680 Sep 12 16:50 file
-rw-r--r--@ 1 choeseoyeon  staff  1223364 Sep 12 16:17 lab03-env-setuid-2.pdf
-rw-r--r--@ 1 choeseoyeon  staff   429653 Sep 12 15:20 lab03-env-setuid.pdf
-rw-r--r--@ 1 choeseoyeon  staff      180 Sep 12 05:39 myenv.c
-rwxr-xr-x  1 choeseoyeon  staff    33184 Sep 12 16:31 myprintenv
-rw-r--r--@ 1 choeseoyeon  staff      571 Sep 12 16:50 myprintenv.c
(base) choeseoyeon@dyn-169-226-220-44 lab03-main %
```

STEP2:

Now comment out the printenv() statement in the child process case (Line ①), and uncomment the printenv() statement in the parent process case (Line ②). Compile and run the code again, and describe your observation. Save the output in another file.

(I will save the new result in file2).

```
_=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main/./a.out
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ./a.out > file2
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ./file2
zsh: permission denied: ./file2
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % cat file2
__CFBundleIdentifier=com.apple.Terminal
TMPDIR=/var/folders/37/rntj73sd0y96f9441_38454w0000gn/T/
XPC_FLAGS=0x0
LaunchInstanceID=22431546-C138-4E73-A4A6-1EBC5D481BA2
TERM=xterm-256color
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.2w8xHHO1Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=0
TERM_PROGRAM=Apple_Terminal
TERM_PROGRAM_VERSION=447
TERM_SESSION_ID=799C7D92-7ECB-4585-BADE-CBBA72606CD9
SHELL=/bin/zsh
HOME=/Users/choeseoyeon
LOGNAME=choeseoyeon
USER=choeseoyeon
PATH=/Users/choeseoyeon/anaconda3/bin:/Users/choeseoyeon/anaconda3/condabin:/usr/local/bin:/usr/lo
cal/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framewo
rk/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/us
r/local/share/dotnet:~/.dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptexd/c
odex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/us
r/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/Users/cho
eseoyeon/.zsh/misc
SHLVL=1
PWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main
OLDPWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3
HOMEBREW_PREFIX=/usr/local
HOMEBREW_CELLAR=/usr/local/Cellar
HOMEBREW_REPOSITORY=/usr/local/Homebrew
MANPATH=/usr/local/share/man::
INFOPATH=/usr/local/share/info:
CONDA_EXE=/Users/choeseoyeon/anaconda3/bin/conda
_CE_M=
_CE_CONDA=
CONDA_PYTHON_EXE=/Users/choeseoyeon/anaconda3/bin/python
CONDA_SHLVL=1
CONDA_PREFIX=/Users/choeseoyeon/anaconda3
CONDA_DEFAULT_ENV=base
CONDA_PROMPT_MODIFIER=(base)
LANG=en_US.UTF-8
_=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main/./a.out
(base) choeseoyeon@dyn-169-226-220-44 lab03-main %
```

// I saved a new output in file2 and used cat command line to display.

```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ls -l
total 3504
-rw-r--r--@ 1 choeseoyeon  staff     1532 Sep 12 05:39 README.md
-rwxr-xr-x  1 choeseoyeon  staff    33184 Sep 12 16:52 a.out
-rwxr-xr-x  1 choeseoyeon  staff    33648 Sep 12 12:30 cap_leak
-rw-r--r--@ 1 choeseoyeon  staff      761 Sep 12 15:13 cap_leak.c
drwxr-xr-x@ 3 choeseoyeon  staff       96 Sep 12 12:30 cap_leak.dSYM
-rw-r--r--@ 1 choeseoyeon  staff      471 Sep 12 05:39 catall.c
-rw-r--r--  1 choeseoyeon  staff     1680 Sep 12 16:50 file
-rw-r--r--  1 choeseoyeon  staff     1680 Sep 12 16:52 file2
-rw-r--r--@ 1 choeseoyeon  staff  1223364 Sep 12 16:17 lab03-env-setuid-2.pdf
-rw-r--r--@ 1 choeseoyeon  staff   429653 Sep 12 15:20 lab03-env-setuid.pdf
-rw-r--r--@ 1 choeseoyeon  staff      180 Sep 12 05:39 myenv.c
-rwxr-xr-x  1 choeseoyeon  staff    33184 Sep 12 16:31 myprintenv
-rw-r--r--@ 1 choeseoyeon  staff      571 Sep 12 16:52 myprintenv.c
(base) choeseoyeon@dyn-169-226-220-44 lab03-main %
```

// As above, I uncommented the parent part's printenv() statement and ran it and saved output into file2. Also used the command line ls -l to check whether there is a difference of permission type between file and file2, but I don't think I see a difference.

STEP3:

Compare the difference of these two files using the diff command. Please draw your conclusion.

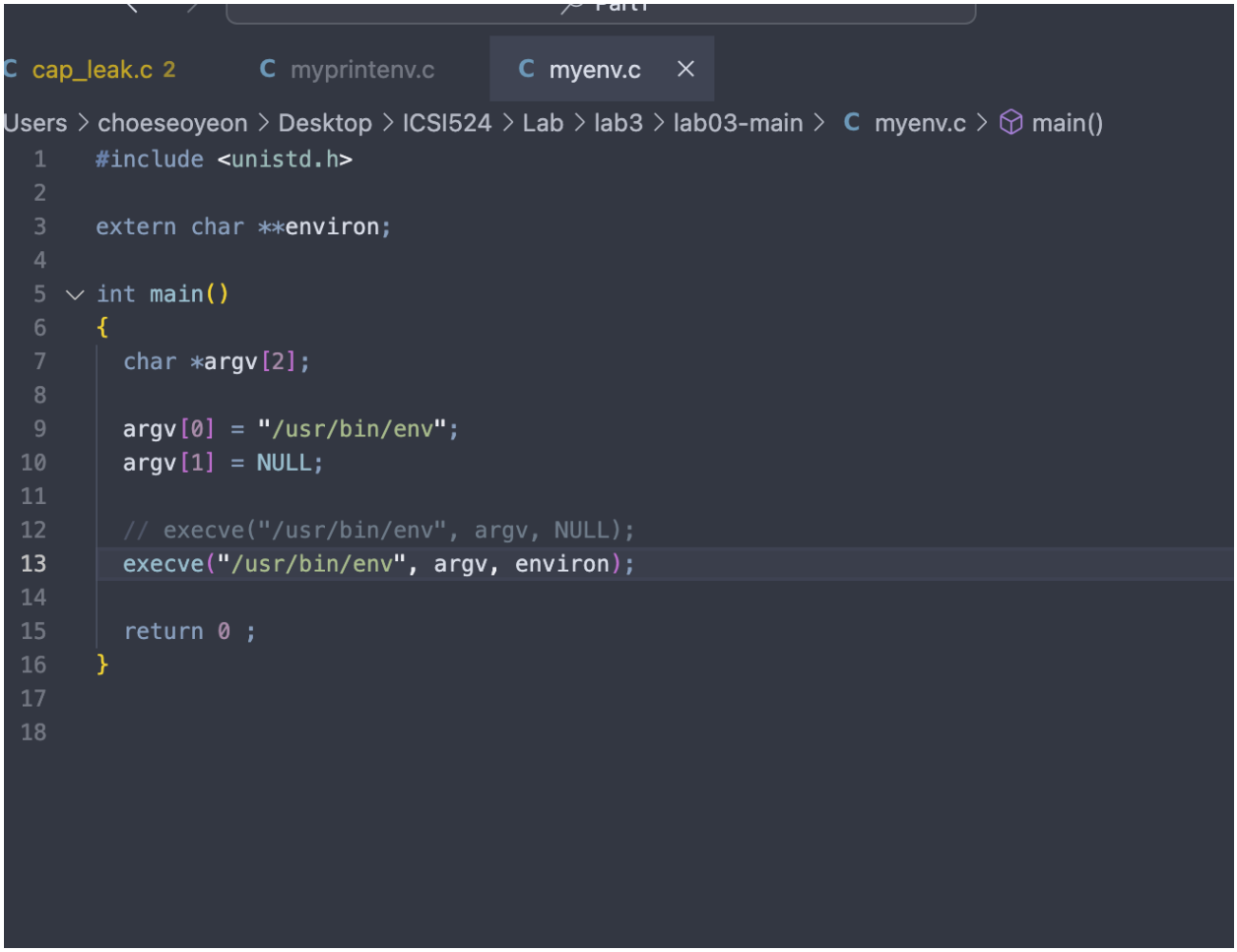
```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % diff file file2
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ls
README.md               cap_leak.dSYM           lab03-env-setuid-2.pdf  myprintenv.c
a.out                   catall.c                lab03-env-setuid.pdf
cap_leak                file                    myenv.c
cap_leak.c              file2                   myprintenv
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % diff file file2
(base) choeseoyeon@dyn-169-226-220-44 lab03-main %
```

// I used diff [file] [file2] to compare both of the files, but it's not printing out anything so those two files don't have a difference, even if the file is created by the child process environment variable and file2 is created by the parent process environment variable. So both of the process's environment variables are the same.

2.3 Task 3: Environment Variables and execve()

```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % /usr/bin/env
__CFBundleIdentifier=com.apple.Terminal
TMPDIR=/var/folders/37/rntj73sd0y96f9441_38454w0000gn/T/
XPC_FLAGS=0x0
LaunchInstanceID=22431546-C138-4E73-A4A6-1EBC5D481BA2
TERM=xterm-256color
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.2w8xHHO1Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=0
TERM_PROGRAM=Apple_Terminal
TERM_PROGRAM_VERSION=447
TERM_SESSION_ID=799C7D92-7ECB-4585-BADE-CBBA72606CD9
SHELL=/bin/zsh
HOME=/Users/choeseoyeon
LOGNAME=choeseoyeon
USER=choeseoyeon
PATH=/Users/choeseoyeon/anaconda3/bin:/Users/choeseoyeon/anaconda3/condabin:/usr/local/bin:/usr/lo
cal/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framewo
rk/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/us
r/local/share/dotnet:~/.dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptexd/c
odex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/us
r/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/Users/cho
eseoyeon/.zsh/misc
SHLVL=1
PWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main
OLDPWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3
HOMEBREW_PREFIX=/usr/local
HOMEBREW_CELLAR=/usr/local/Cellar
HOMEBREW_REPOSITORY=/usr/local/Homebrew
MANPATH=/usr/local/share/man::
INFOPATH=/usr/local/share/info:
CONDA_EXE=/Users/choeseoyeon/anaconda3/bin/conda
_CE_M=
_CE_CONDA=
CONDA_PYTHON_EXE=/Users/choeseoyeon/anaconda3/bin/python
CONDA_SHLVL=1
CONDA_PREFIX=/Users/choeseoyeon/anaconda3
CONDA_DEFAULT_ENV=base
CONDA_PROMPT_MODIFIER=(base)
LANG=en_US.UTF-8
_=/usr/bin/env
(base) choeseoyeon@dyn-169-226-220-44 lab03-main %
```

// Task1: simply call “ /usr/bin/env” to compile the file.



//Task2 : change line 12 as above

```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % /usr/bin/env
__CFBundleIdentifier=com.apple.Terminal
TMPDIR=/var/folders/37/rntj73sd0y96f9441_38454w0000gn/T/
XPC_FLAGS=0x0
LaunchInstanceID=22431546-C138-4E73-A4A6-1EBC5D481BA2
TERM=xterm-256color
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.2w8xHHO1Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=0
TERM_PROGRAM=Apple_Terminal
TERM_PROGRAM_VERSION=447
TERM_SESSION_ID=799C7D92-7ECB-4585-BADE-CBBA72606CD9
SHELL=/bin/zsh
HOME=/Users/choeseoyeon
LOGNAME=choeseoyeon
USER=choeseoyeon
PATH=/Users/choeseoyeon/anaconda3/bin:/Users/choeseoyeon/anaconda3/condabin:/usr/local/bin:/usr/lo
cal/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framewo
rk/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/us
r/local/share/dotnet:~/.dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptexd/c
odex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/us
r/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/Users/cho
eseoyeon/.zsh/misc
SHLVL=1
PWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main
OLDPWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3
HOMEBREW_PREFIX=/usr/local
HOMEBREW_CELLAR=/usr/local/Cellar
HOMEBREW_REPOSITORY=/usr/local/Homebrew
MANPATH=/usr/local/share/man::
INFOPATH=/usr/local/share/info:
CONDA_EXE=/Users/choeseoyeon/anaconda3/bin/conda
_CE_M=
_CE_CONDA=
CONDA_PYTHON_EXE=/Users/choeseoyeon/anaconda3/bin/python
CONDA_SHLVL=1
CONDA_PREFIX=/Users/choeseoyeon/anaconda3
CONDA_DEFAULT_ENV=base
CONDA_PROMPT_MODIFIER=(base)
LANG=en_US.UTF-8
_=/usr/bin/env
(base) choeseoyeon@dyn-169-226-220-44 lab03-main %
```

So if I use the execve() system call method and create a process, then current process memory will be covered/overwritten by the newly created program. So in this process, existing environmental variables will be covered/disappear and new environmental variables created by the execve() system call will be passed.

Also if I uncomment line 13 (which has NULL for the environment array list) and run with "/usr/bin/env". It won't print out environment variables because line 13 setted the environment array list as NULL.

If I comment line 13 and use line 14 arguments, since the environment array list part is not NULL, the current process will get environment variables. So with line 14, the current process can inherit the process's environment variables.

2.4 Task 4: Environment Variables and system()

```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % gcc task4.c -o task4
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ls
README.md               cap_leak.dSYM           lab03-env-setuid-2.pdf  myprintenv
a.out                   catall.c                lab03-env-setuid.pdf    myprintenv.c
cap_leak                file                    myenv                   task4
cap_leak.c              file2                   myenv.c                 task4.c
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ./task4
MANPATH=/usr/local/share/man::
TERM_PROGRAM=Apple_Terminal
SHELL=/bin/zsh
TERM=xterm-256color
HOMEBREW_REPOSITORY=/usr/local/Homebrew
TMPDIR=/var/folders/37/rntj73sd0y96f9441_38454w0000gn/T/
CONDA_SHLVL=1
CONDA_PROMPT_MODIFIER=(base)
TERM_PROGRAM_VERSION=447
TERM_SESSION_ID=799C7D92-7ECB-4585-BADE-CBBA72606CD9
USER=choeseoyeon
CONDA_EXE=/Users/choeseoyeon/anaconda3/bin/conda
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.2w8xHHO1Ef/Listeners
_CE_CONDA=
PATH=/Users/choeseoyeon/anaconda3/bin:/Users/choeseoyeon/anaconda3/condabin:/usr/local/bin:/usr/lo
cal/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framewo
rk/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/us
r/local/share/dotnet:~/.dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptexd/c
odex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/us
r/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/Users/cho
eseoyeon/.zsh/misc
_=/usr/bin/env
LaunchInstanceID=22431546-C138-4E73-A4A6-1EBC5D481BA2
CONDA_PREFIX=/Users/choeseoyeon/anaconda3
__CFBundleIdentifier=com.apple.Terminal
PWD=/Users/choeseoyeon/Desktop/ICSI524/Lab/lab3/lab03-main
LANG=en_US.UTF-8
XPC_FLAGS=0x0
_CE_M=
XPC_SERVICE_NAME=0
SHLVL=2
HOME=/Users/choeseoyeon
HOMEBREW_PREFIX=/usr/local
CONDA_PYTHON_EXE=/Users/choeseoyeon/anaconda3/bin/python
LOGNAME=choeseoyeon
CONDA_DEFAULT_ENV=base
INFOPATH=/usr/local/share/info:
HOMEBREW_CELLAR=/usr/local/Cellar
SECURITYSESSIONID=18723
(base) choeseoyeon@dyn-169-226-220-44 lab03-main %
```

// I created task4.c file that contains task4 code and run it by using gcc command line.

For System() function, this function calls the program and prints out the values that pass through the perimeter(). So, the system() function will execute a shell command. Also System() is the same as the "/bin/sh -c command" command line.

fork() and create child process -> this process do /bin/sh and run excel(), execl() takes environment variables , so environment variable inherientes current process's environment

variables. However, it doesn't mean that shell variables will be the same as environment variables.

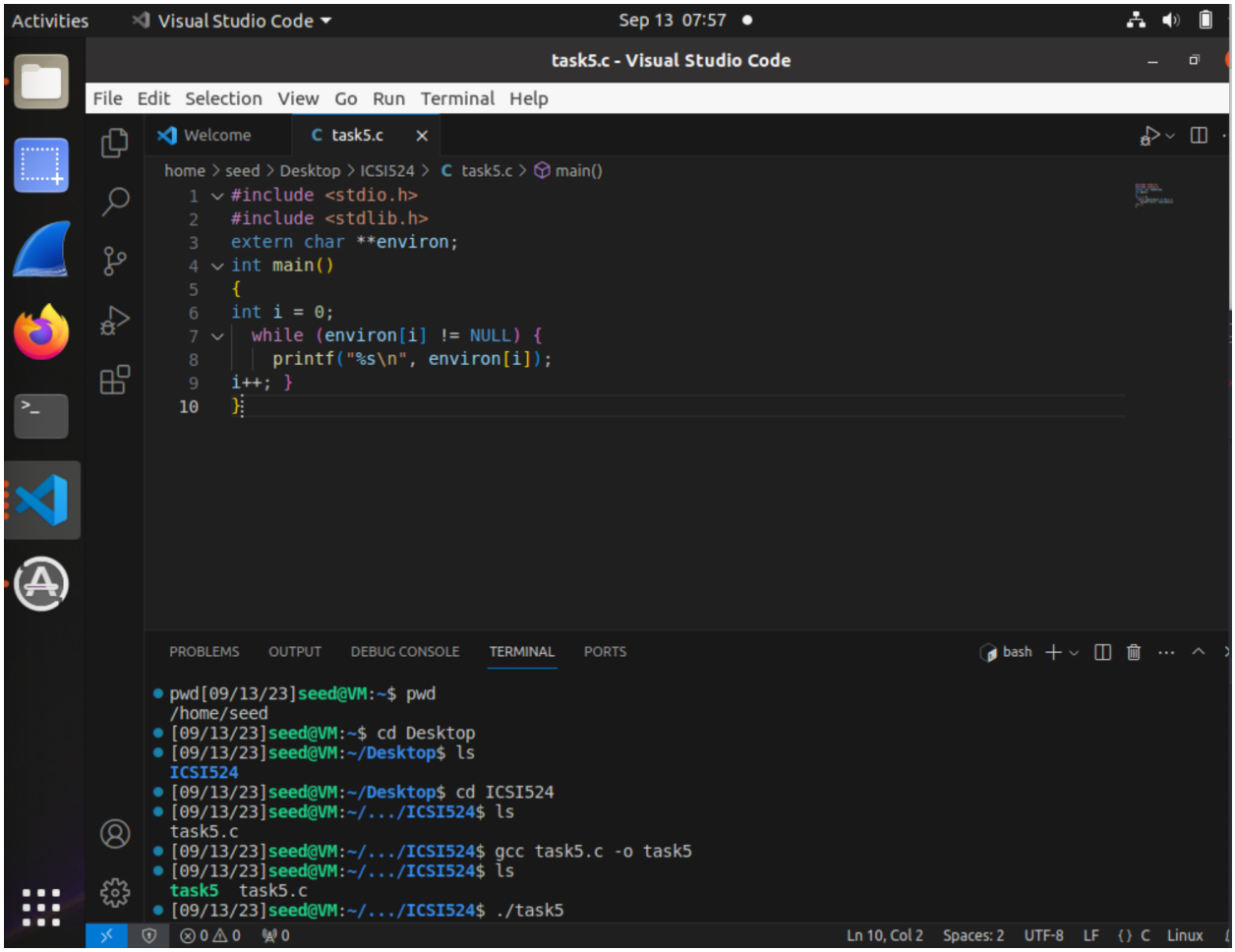
For the /usr/bin/env command line, this command line is used to print out environment variables. So this command line will return a list of environment variables.

2.5 Task 5: Environment Variable and Set-UID Programs

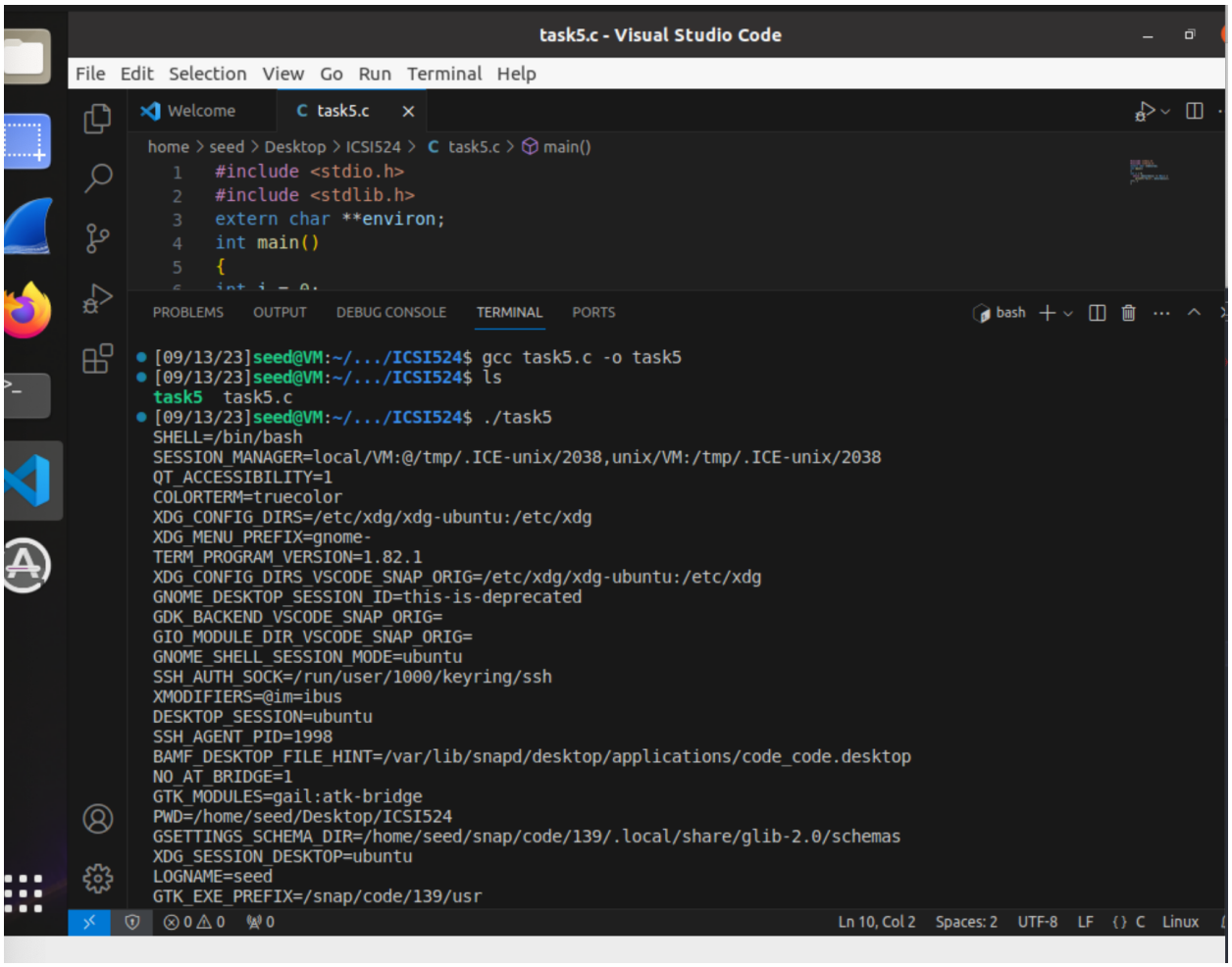
STEP1: I write down give code with named file task5.c

-> I did the Visual studio code.

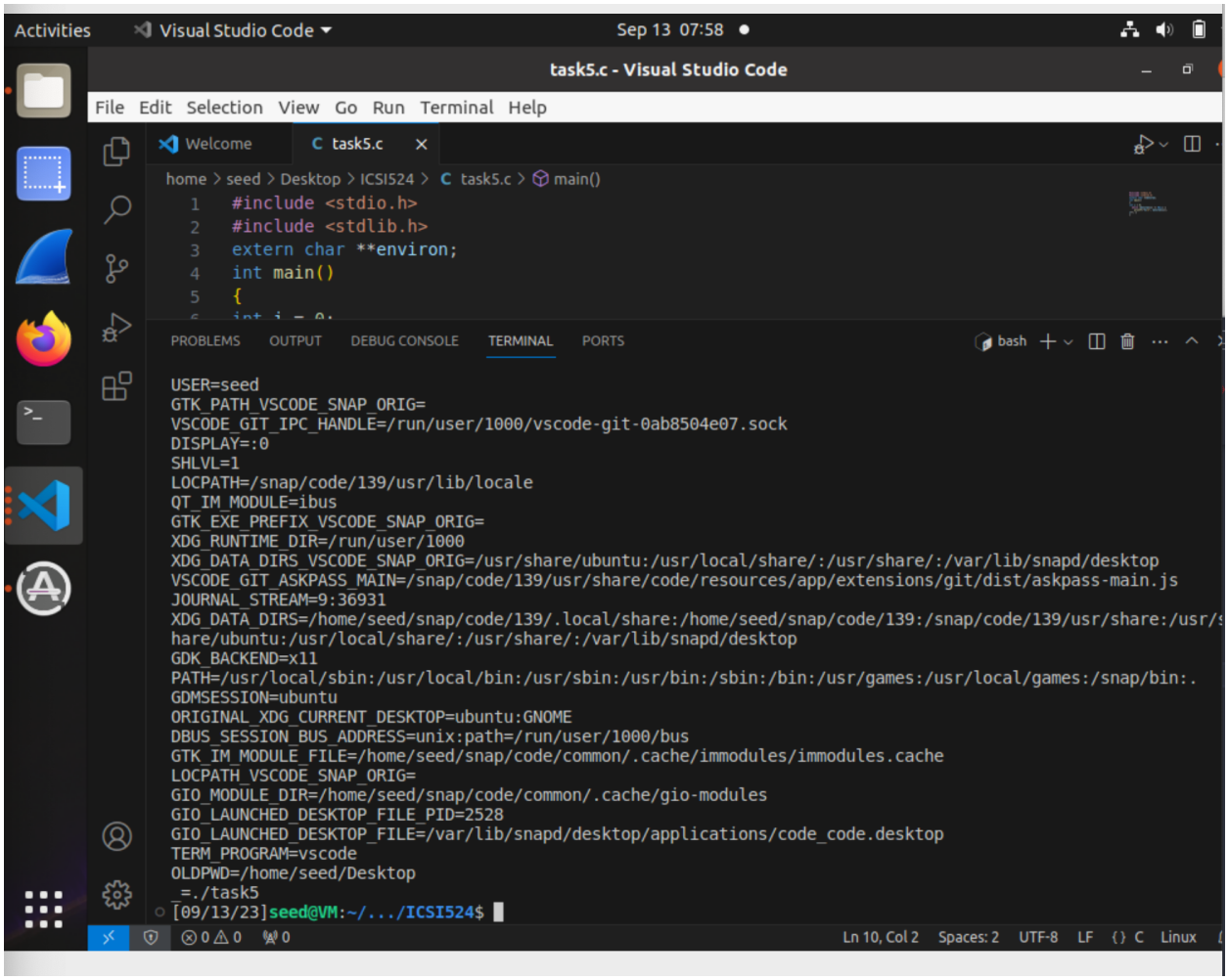
STEP2: Compile the above program, change its ownership to root, and make it a Set-UID program.



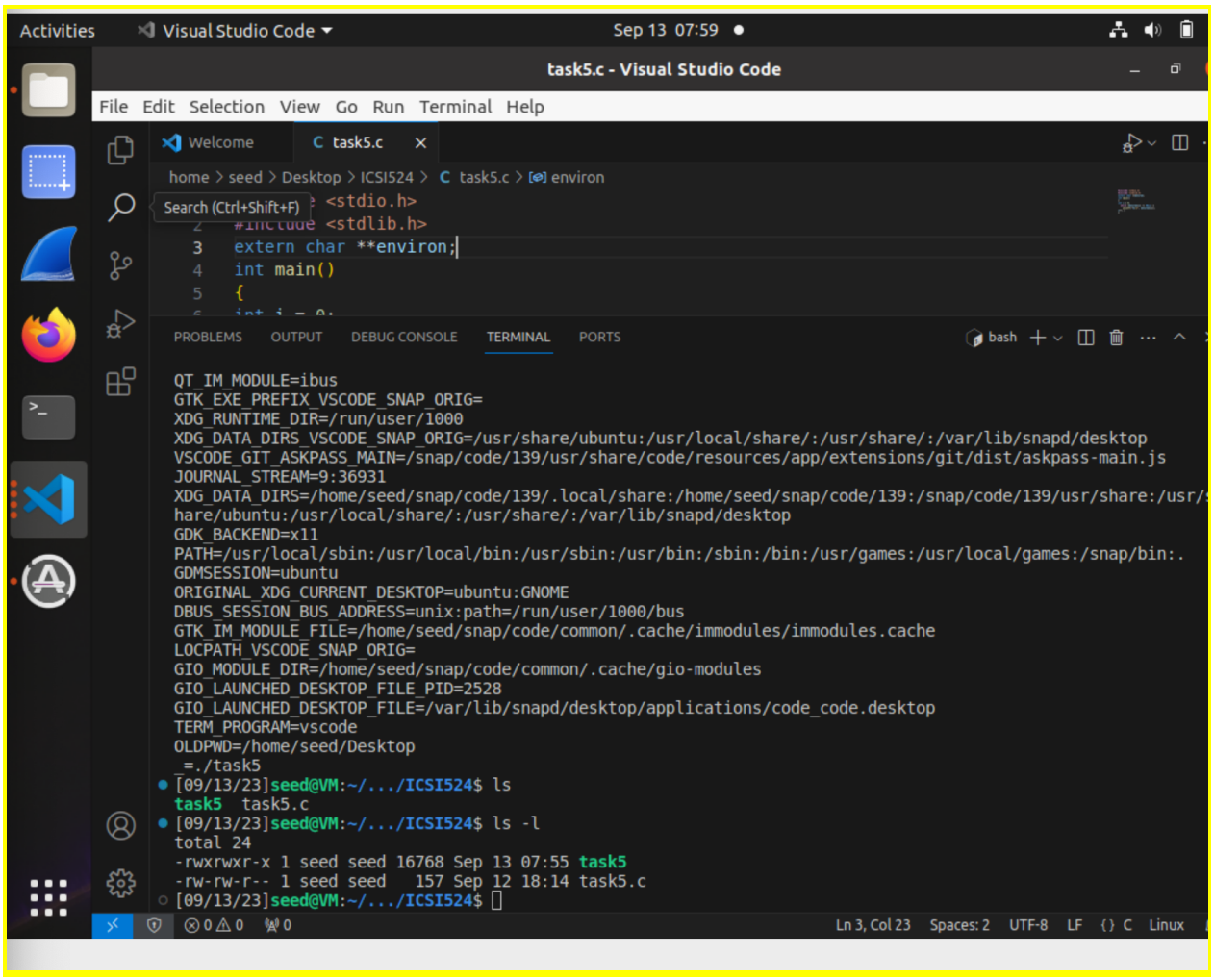
// I created a name called task5.c and compiled the file using gcc command line.



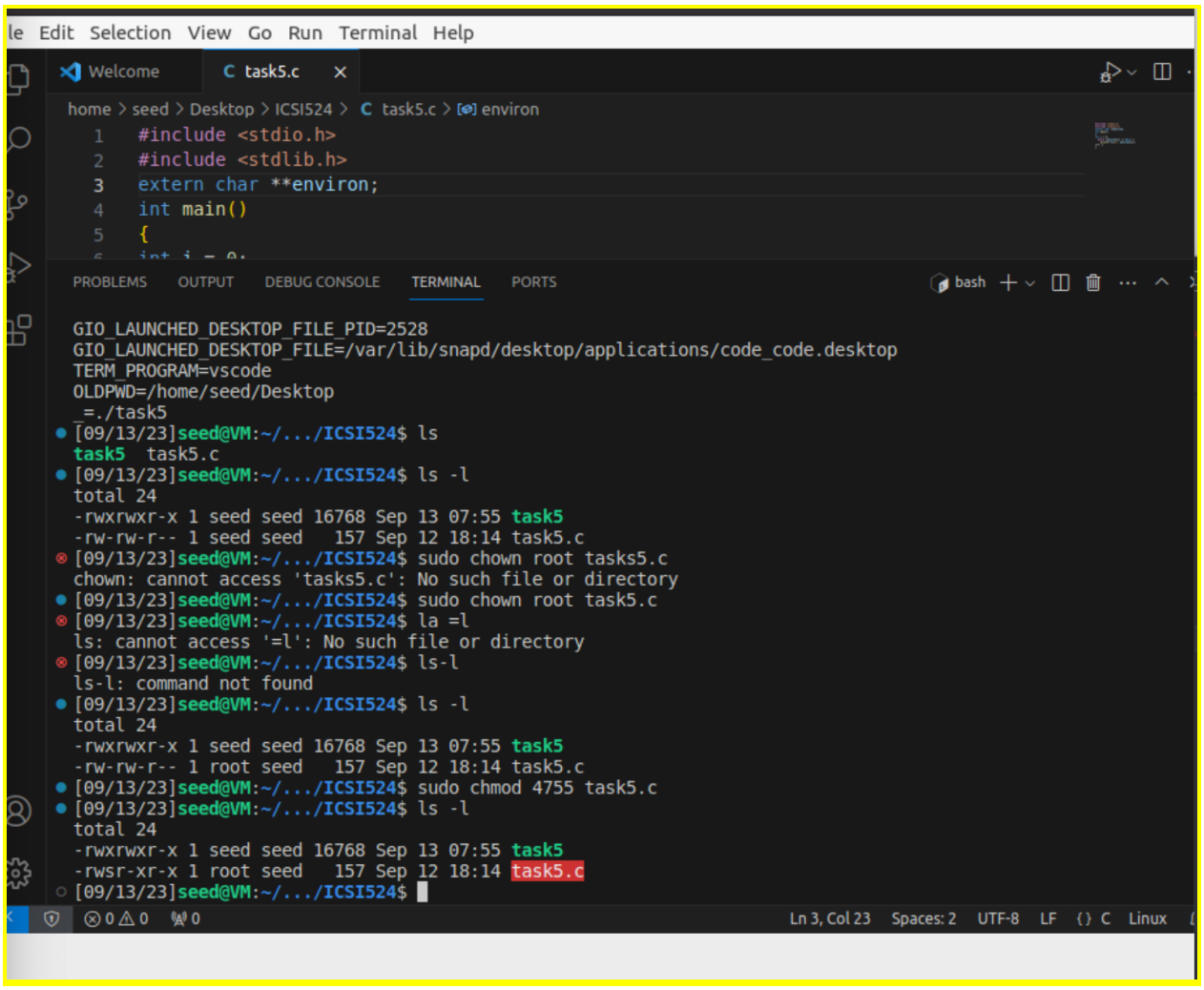
// I ran the task5 file to see information about the program.



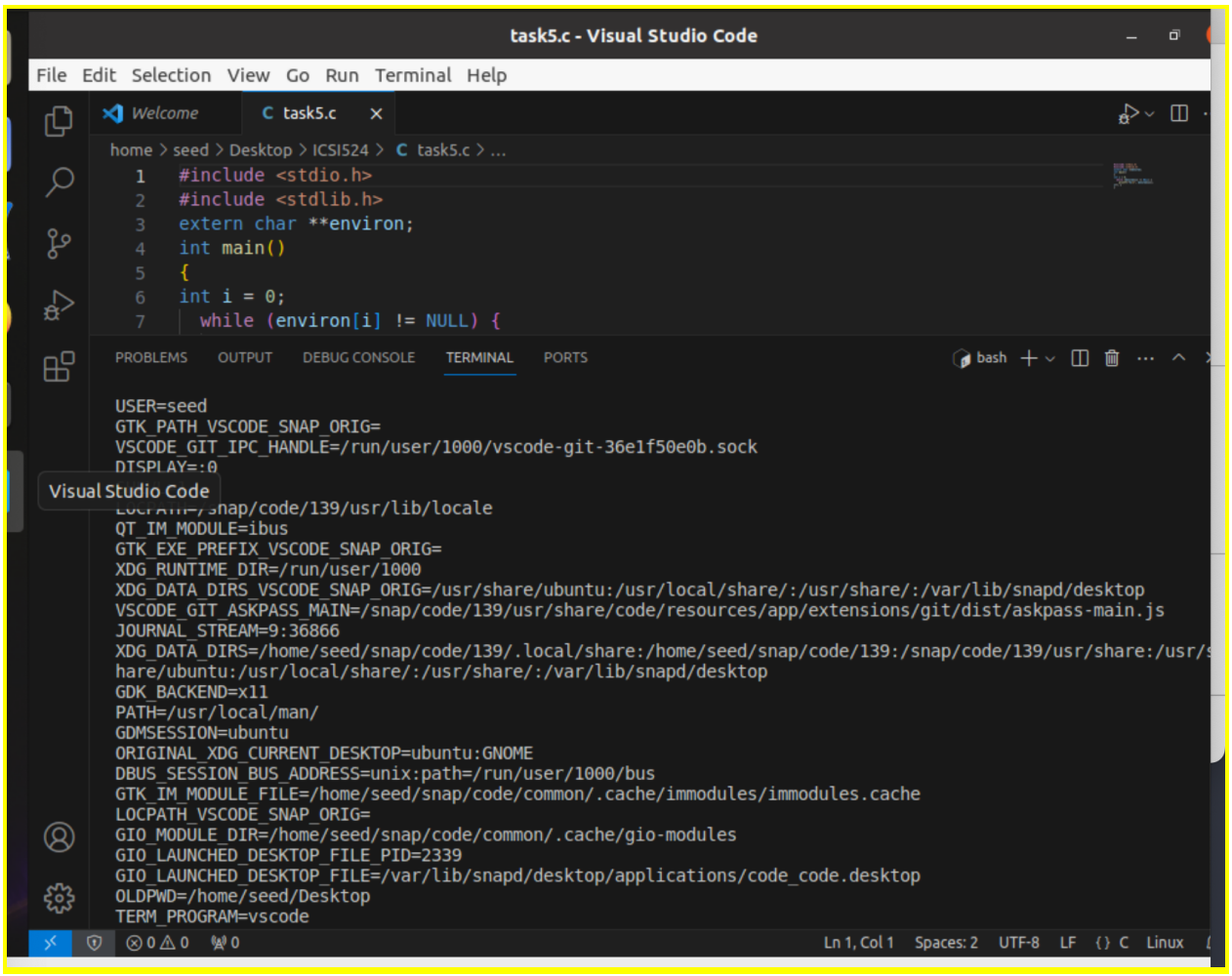
// I will check permission and owner of task5.c



// I will use the given 2 command lines to change ownership permission of the file to the root.

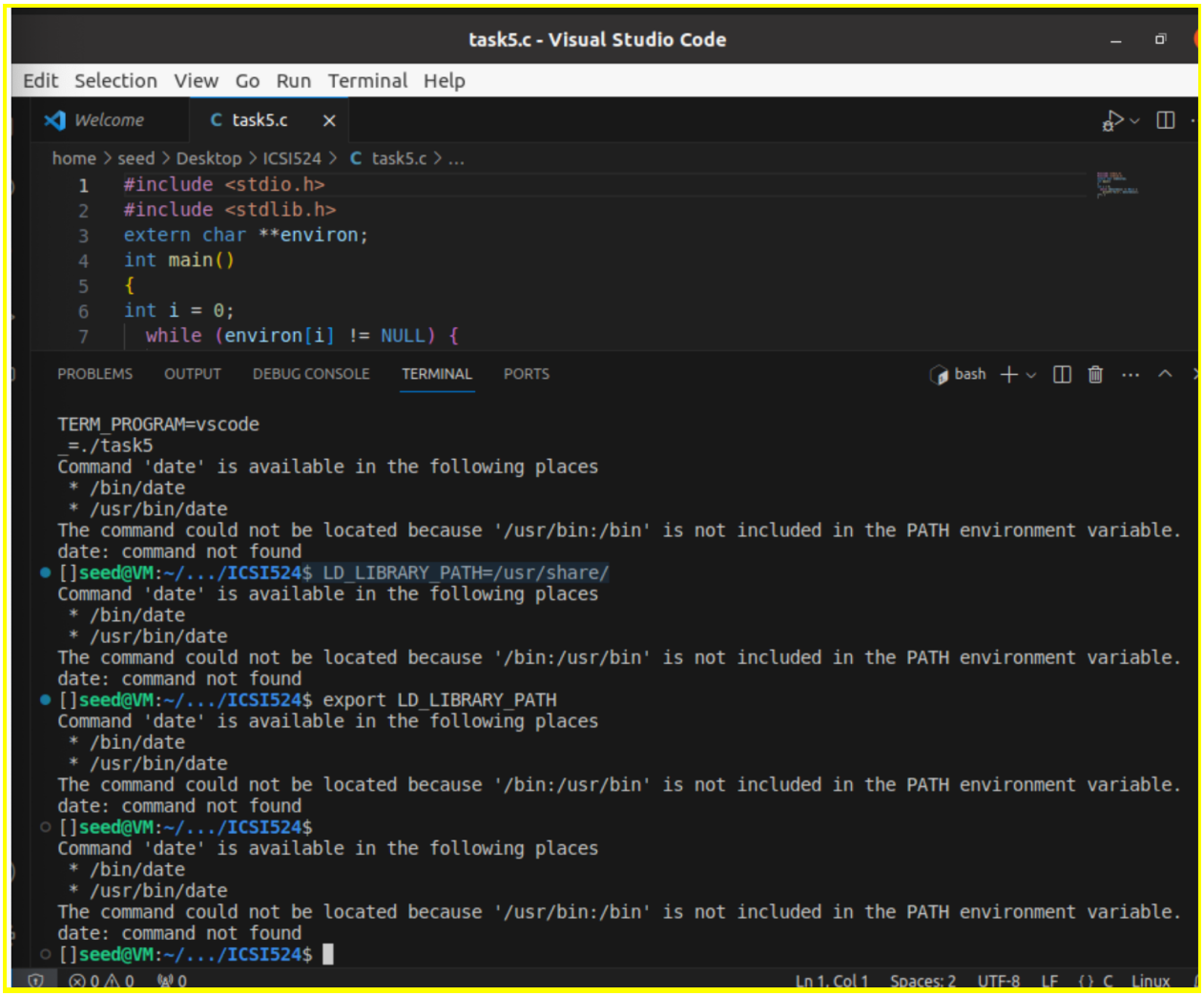


I typed PATH=”/usr/local/man/” and exported the path and it changed task5 files PATH.

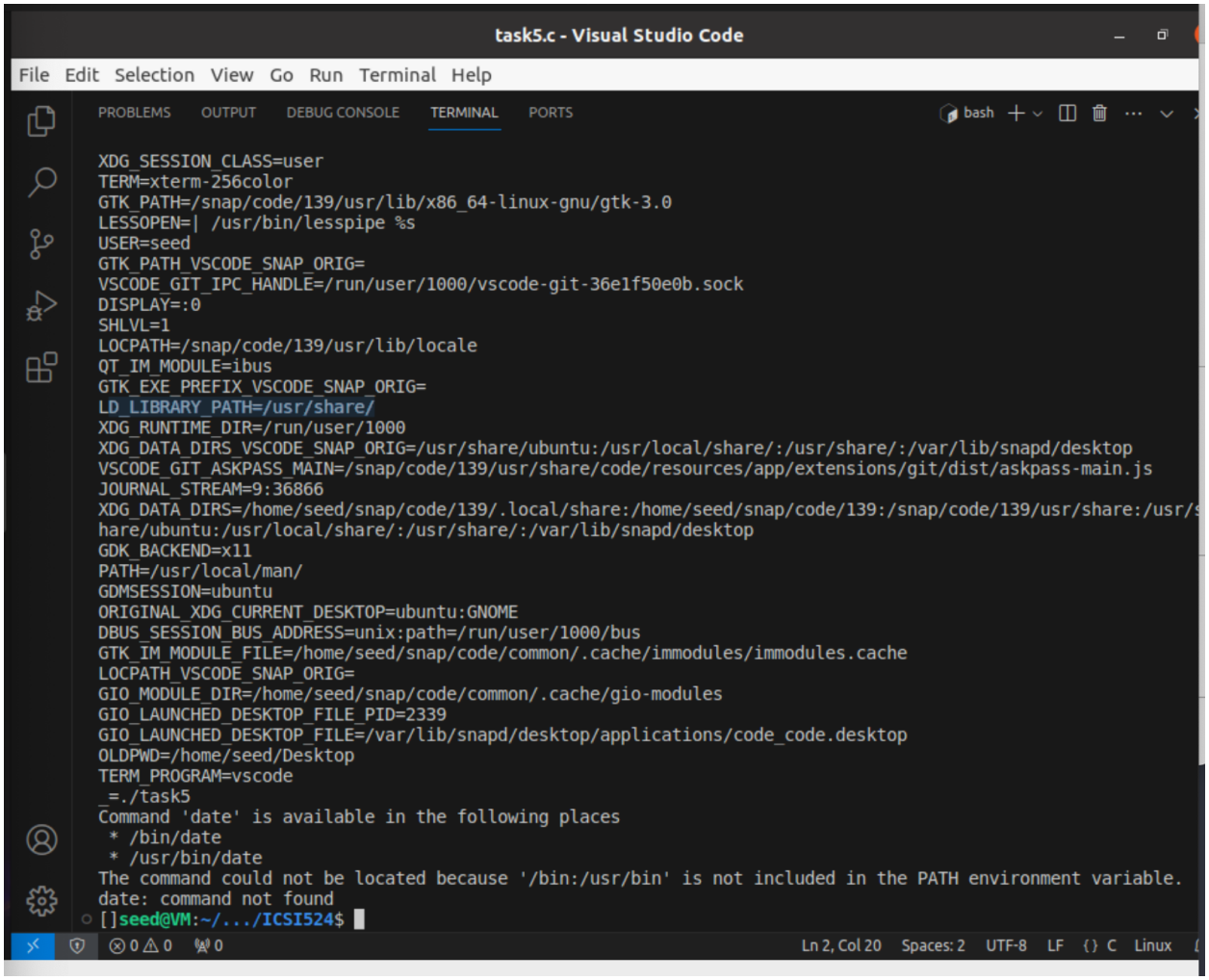


// This shows that PATH has been changed to “ /usr/local/man/ “

// CHANGE LD_LIBRARY_PATH AND EXPORT



// result



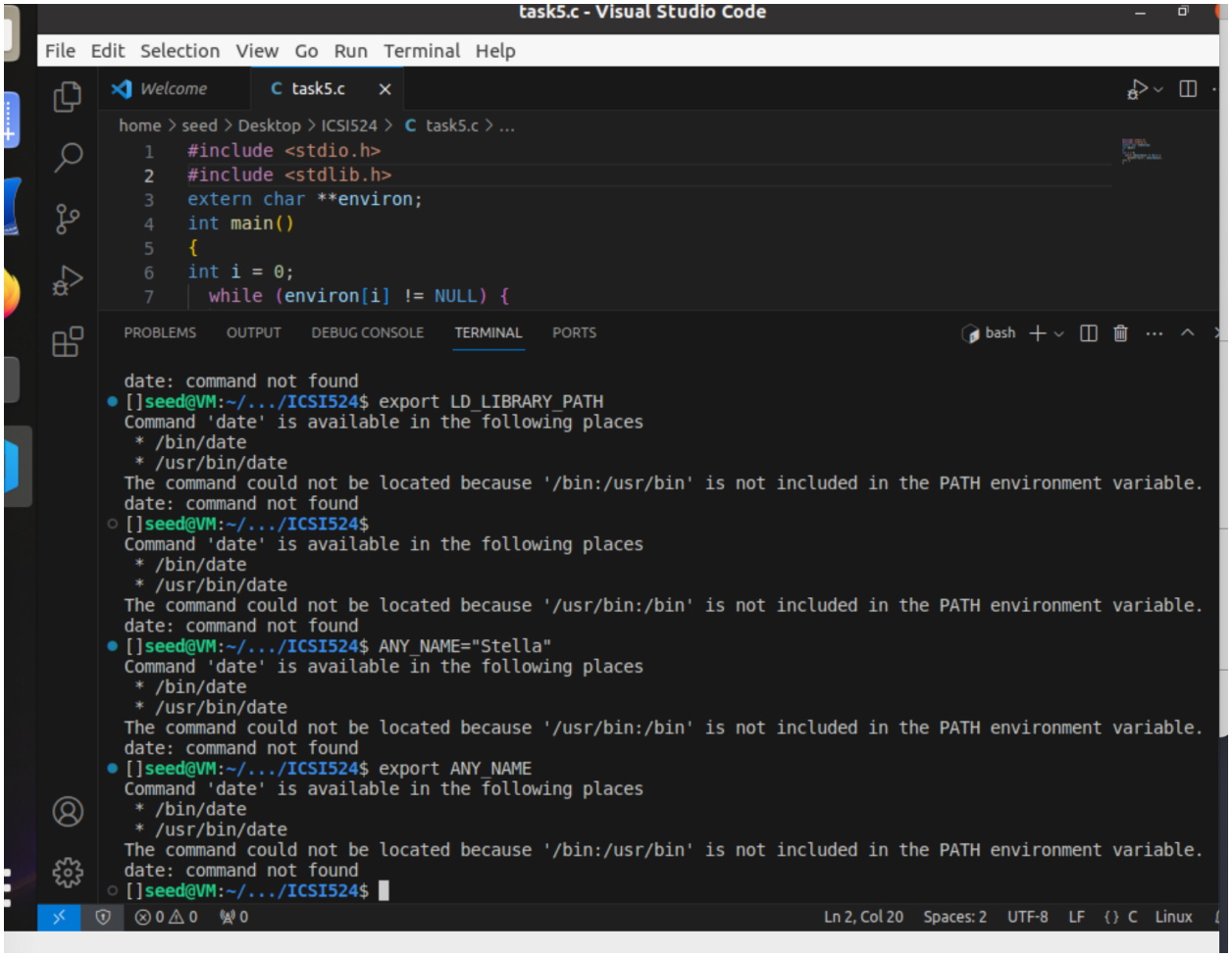
// Above result shows that LD_LIBRARY_PATH has been changed.

// SET ANY_NAME as Stella and Export it.

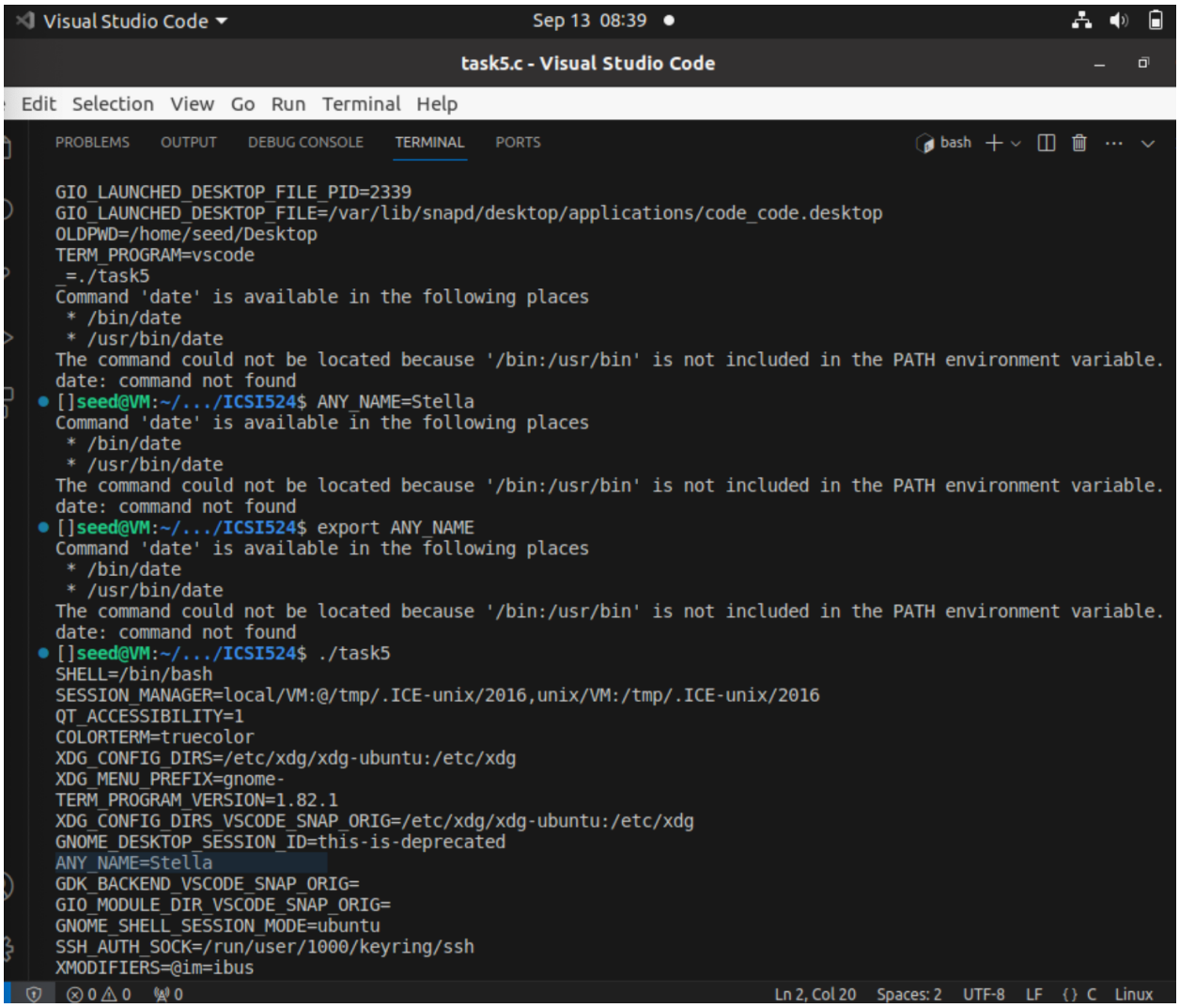
ex) ANY_NAME = Stella

export ANY_NAME

This will update my name as Stella.



//Result will show that ANY_NAME has been changed.



// This shows that ANY_NAME has been changed to Stella.

We are changing the ownership permission of the file to root. So that the file is now under the user root. If any non-root user tries to access the file, a new process will be forked with root permission and the file will be executed under root permission in that process.

Next, we are trying to set the environmental variable using export command. The export command sets the environmental variable for all the processes in the system. So even the root users will be accessing the new environmental variables set using the export command. That is why we are getting the value that we have set.

So, when shell executes, it creates a child process using fork() and then it uses exec() . This covers the current process and in this process, the current process's environment variable will be inherited and the Set-UID program also gets affected.

2.6 Task 6: The PATH Environment Variable and Set-UID Programs

Step1 :
// I attached the screenshot below.

Step2:
// change owner to root and give set-uid

Can you get this Set-UID program to run your own malicious code, instead of /bin/ls? If you can, is your malicious code running with the root privilege? Describe and explain your observations.

Answer is Yes we can. If the path is not specified, the Set-UID program can get a security attack by malicious code. I will proof this as below.

// This is from a textbook. The textbook talks about how we should link bin/zsh with bin/sh and then unlink when we finish this task.

ex) before we should link bin/zsh with bin/sh : $ sudo ln -sf/bin/zsh /bin/sh
After we finished think task, we have to unlink, so : $sudo ln -sf/bin/dash /bin/sh

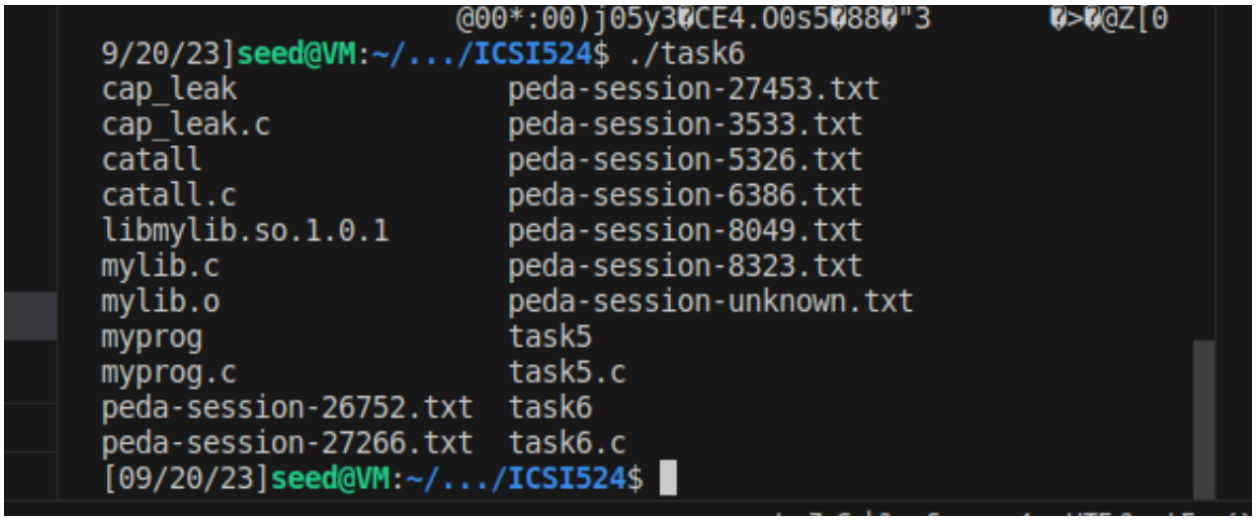
System() method uses system call and it is the same as /bin/sh-c command in the terminal. So it passes the current process's environmental variable s and shell variable that we exported by using export command line into the new process's environment variable.
Also, this will search shell's PATH list, even if we didn't type the full path to search.

So provided command line $ export PATH=/home/seed:$PATH with system() search "ls" in /home/seed/ls instead of /bin/ls.
We can test this by creating an extra malicious file program.

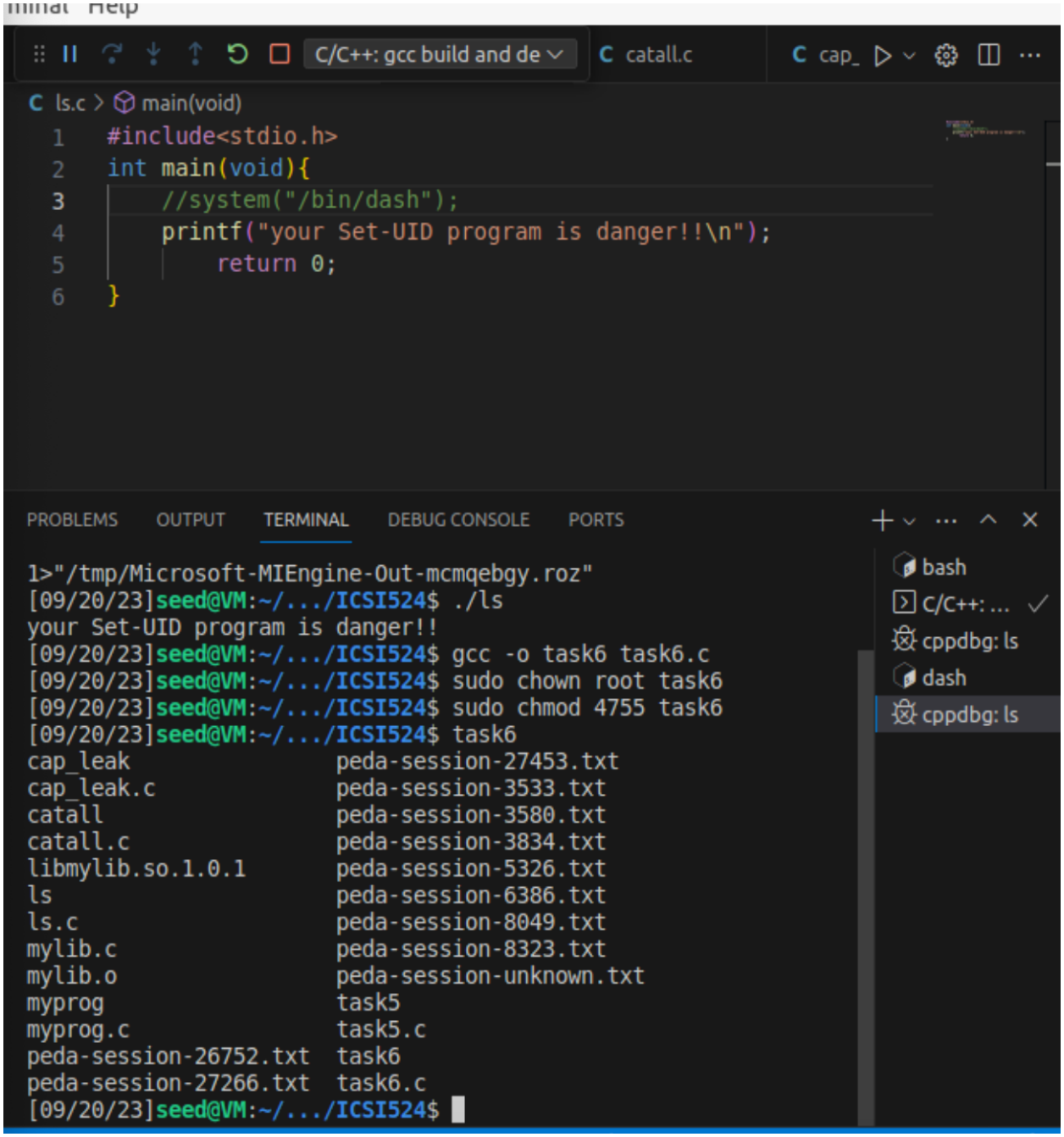
// start this

1. Create task6.c file and compile.

2. Run the task6 by using the ./task6 command line to check whether the task6 program works .

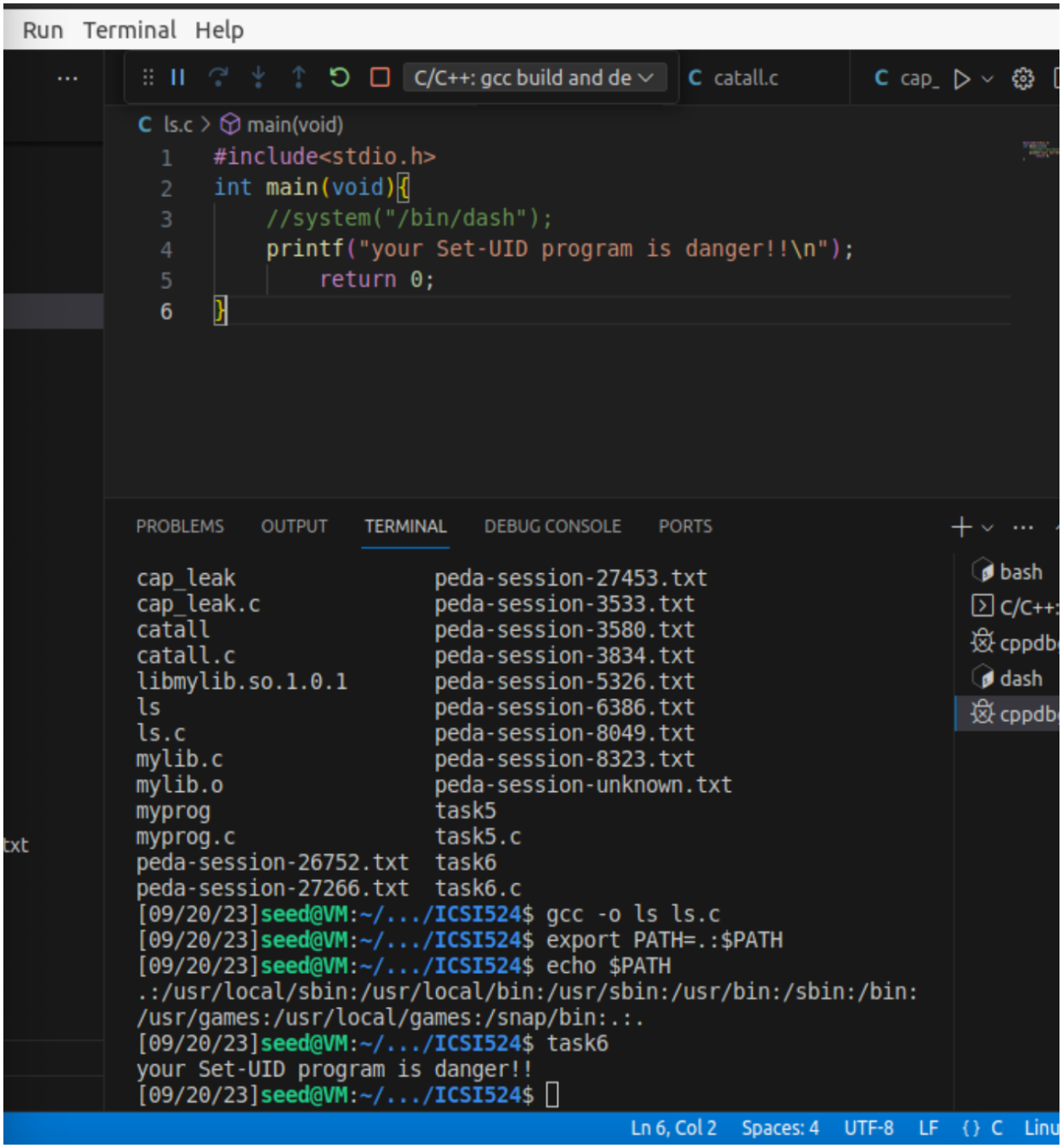


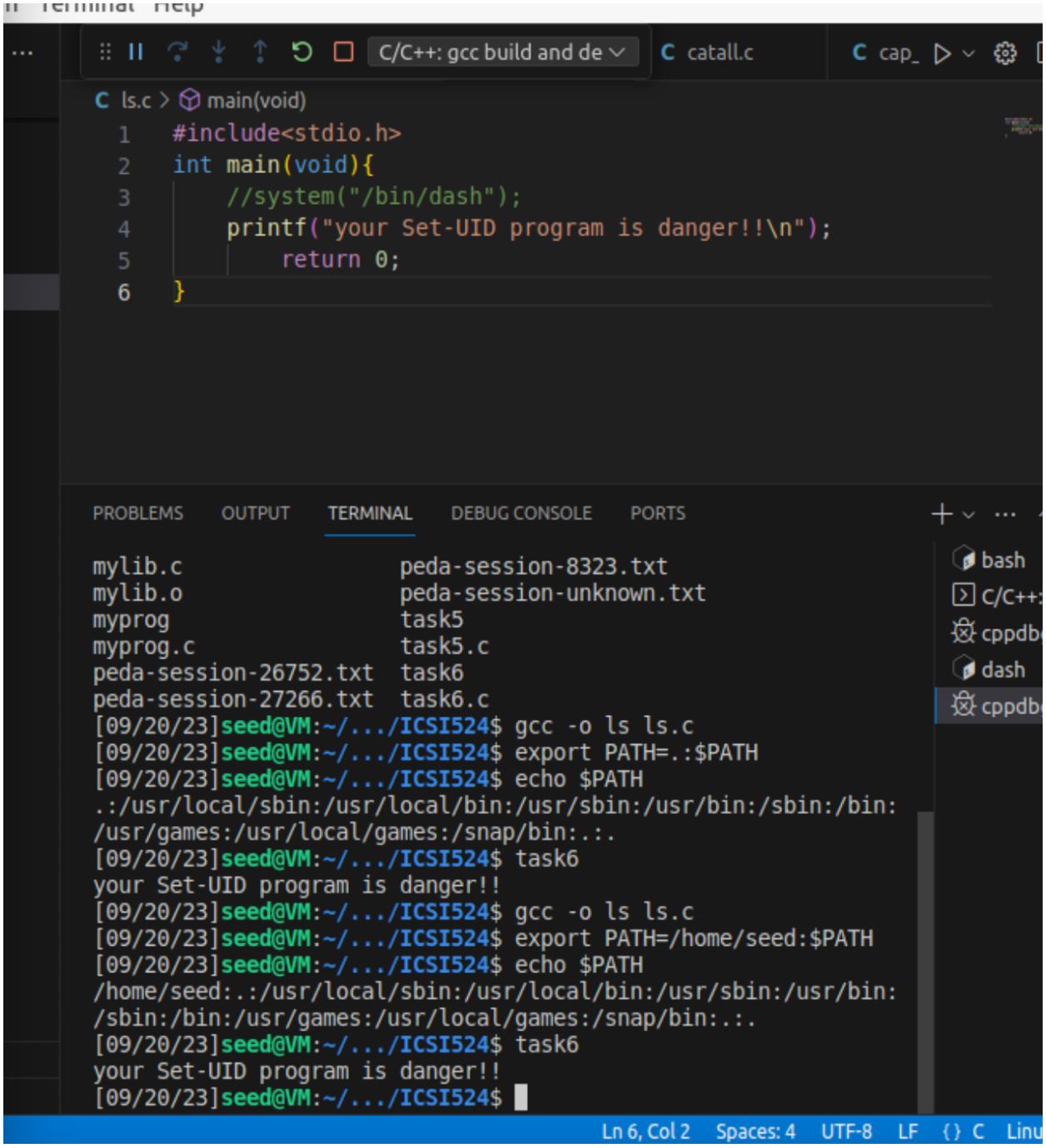
3. Create a malicious program named ls.c and compile the code.



// And then I compile ls.c (malicious program) and then used export line to change the path
I echo $path to show how path changed

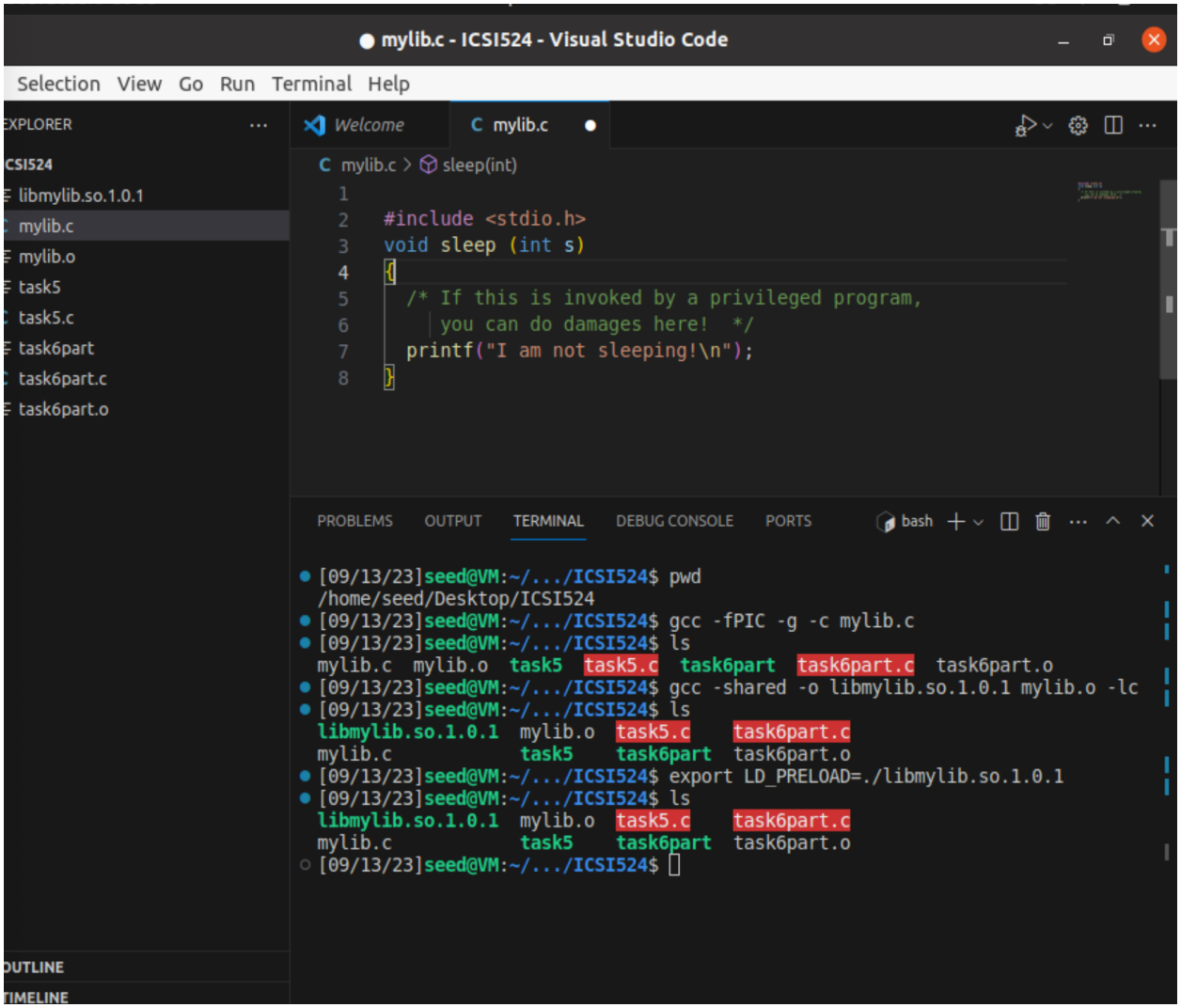
And then I call task6 and this won't function well instead this will get affected by ls.c malicious code and compile malicious code since task6.c changed owner to the root and set-UID.
This shows that we can attack if the path is not specified with previous mode.


Run Terminal Help
C/C++: gcc build and de
catall.c
cap_
ls.c > main(void)
1 #include<stdio.h>
2 int main(void){
3 //system("/bin/dash");
4 printf("your Set-UID program is danger!!\n");
5 return 0;
6 }
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
cap_leak peda-session-27453.txt
cap_leak.c peda-session-3533.txt
catall peda-session-3580.txt
catall.c peda-session-3834.txt
libmylib.so.1.0.1 peda-session-5326.txt
ls peda-session-6386.txt
ls.c peda-session-8049.txt
mylib.c peda-session-8323.txt
mylib.o peda-session-unknown.txt
myprog task5
myprog.c task5.c
peda-session-26752.txt task6
peda-session-27266.txt task6.c
[09/20/23]seed@VM:~/.../ICSI524$ gcc -o ls ls.c
[09/20/23]seed@VM:~/.../ICSI524$ export PATH=.:$PATH
[09/20/23]seed@VM:~/.../ICSI524$ echo $PATH
.:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:
/usr/games:/usr/local/games:/snap/bin:.:.
[09/20/23]seed@VM:~/.../ICSI524$ task6
your Set-UID program is danger!!
[09/20/23]seed@VM:~/.../ICSI524$
bash
C/C++:
cppdb
dash
cppdb
Ln 6, Col 2 Spaces: 4 UTF-8 LF { } C Linu

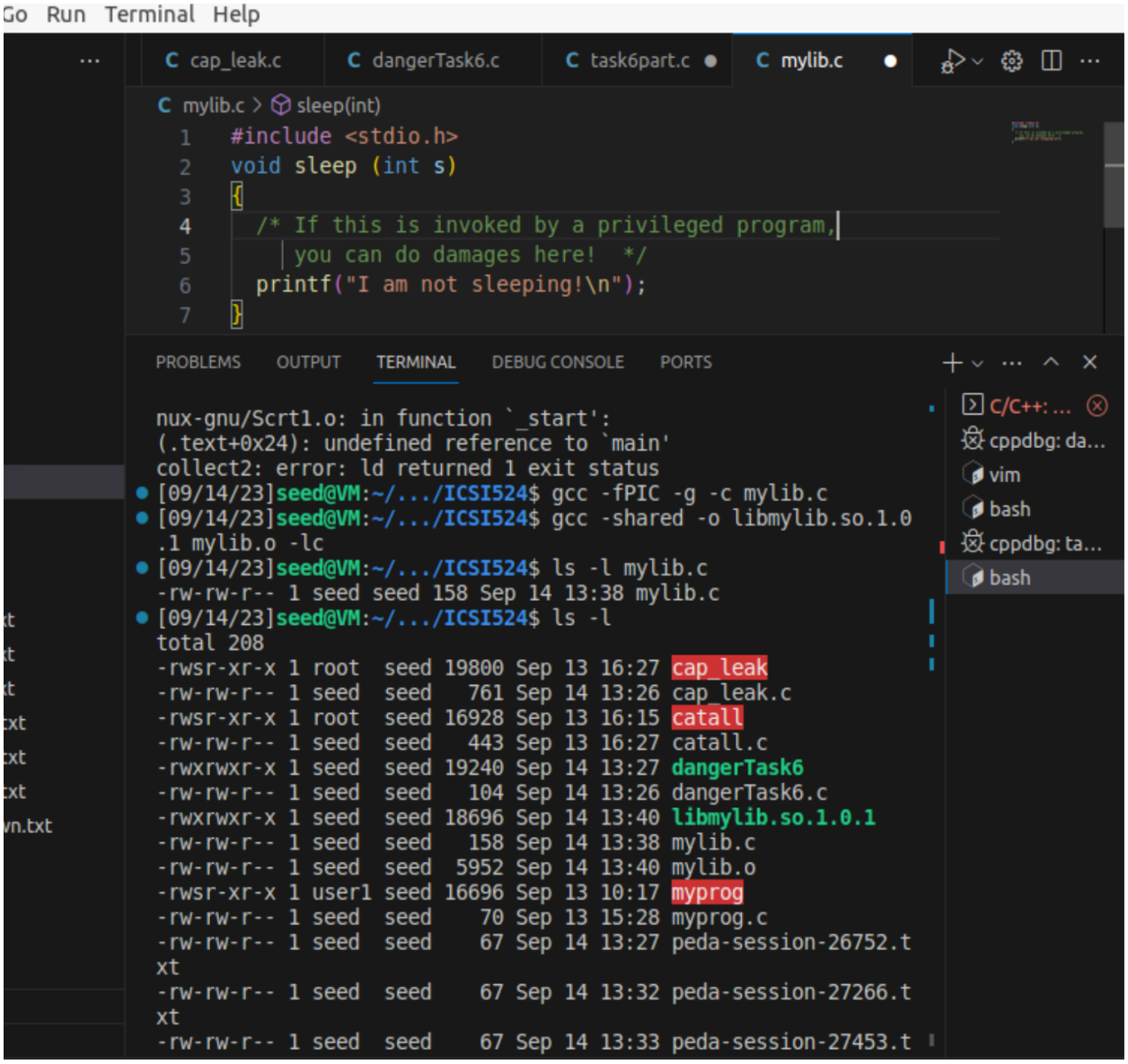


Task 7: The LD_PRELOAD Environment Variable and Set-UID Programs

Step1:

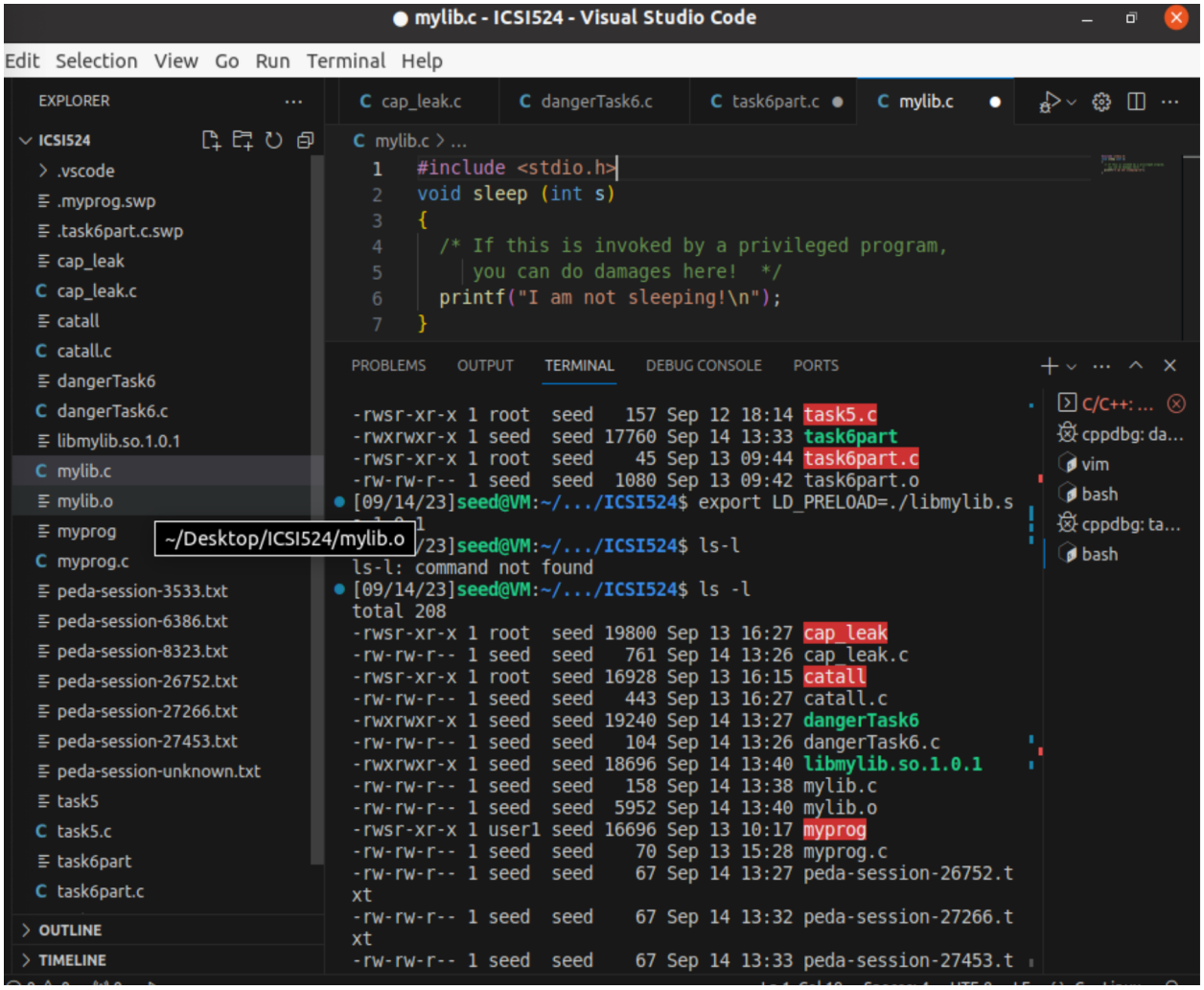


// compile myprog.c



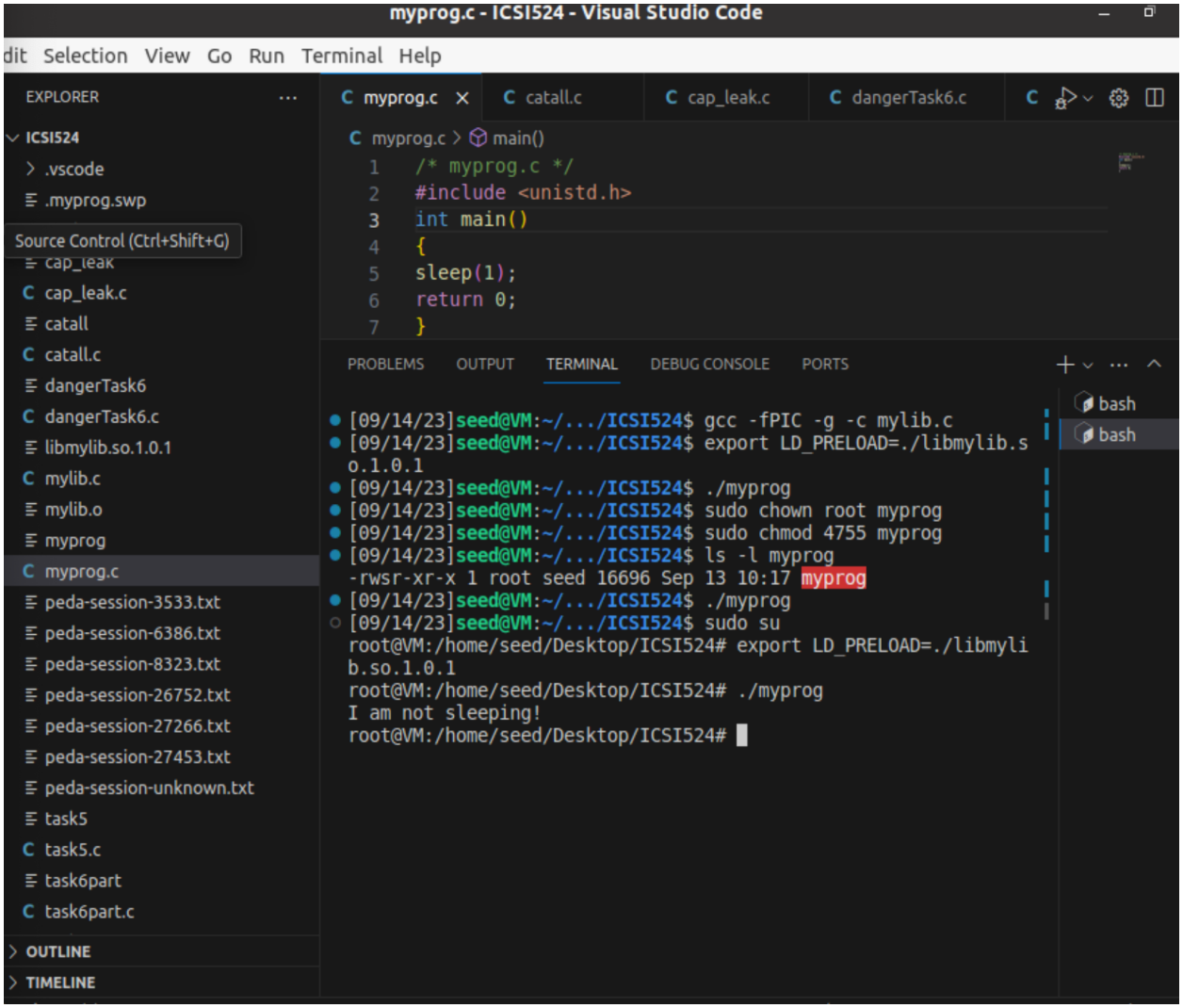
// used two given command line in step2 and run it with mylib.c

//step3 : Now, set the LD_PRELOAD environment variable:



// step4 : Finally,compile the following program myprog, and in the same directory as the above dynamic link library libmylib.so.1.0.1:

2-1,2-2 doesn't print out i am not sleeping, but 2-3 method print out "I am not sleeping"



Like other environment variables, LD_PRELOAD declared as export is also passed as an environment variable when a new program starts.

In Step2-(1), you can see that the redefined sleep() function is executed using our newly created libmylib.so.1.0.1 using LD_PRELOAD.

In Step2-(2), the myprog program was changed to a root program with Set-UID authority. In recent environments, a countermeasure is applied to the dynamic linker (ld.so or ld-linux.so), but if the real ID and effective ID of the process are different,

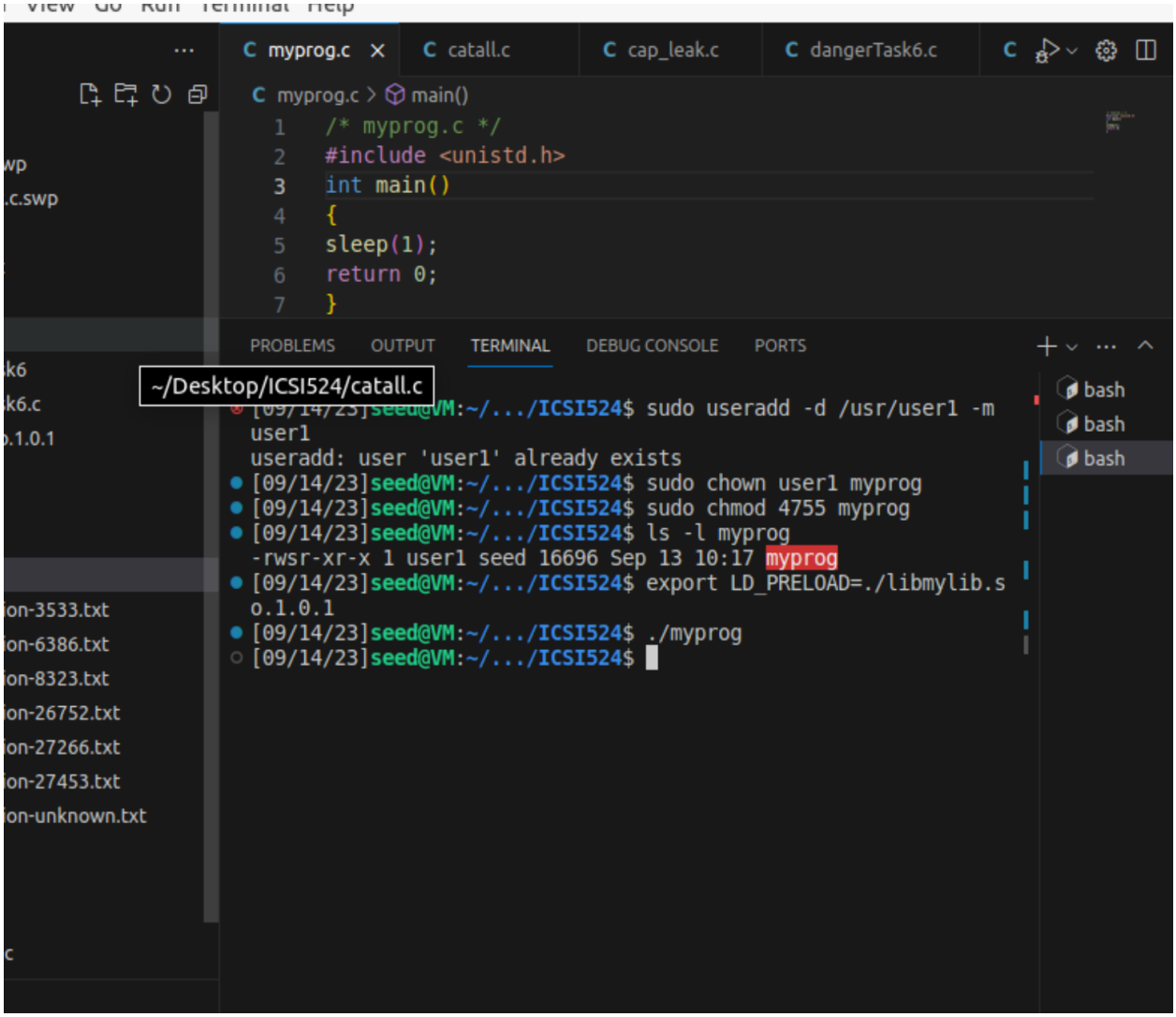
LD_PRELOAD environment variables are ignored. Therefore, you can see that the sleep() function is executed.

-- The real user ID: The user who runs the process

-- The effective user ID: user used for access control

In Step2-(3), you can check that the LD_PRELOAD environment variables are executed without being ignored because the real ID and effective ID of the process are the same.

// Below shows the output of 2-4 options. It doesn't print out " I am not sleeping"



In Step2-(4) as well, since the real ID and effective ID of the process are different, LD_PRELOAD is ignored and the sleep() function is executed.

2.8 Task 8: Invoking External Programs Using system() versus execve()

Step1:
gcc -o catall catall.c
Change owner to root : sudo chown root task8

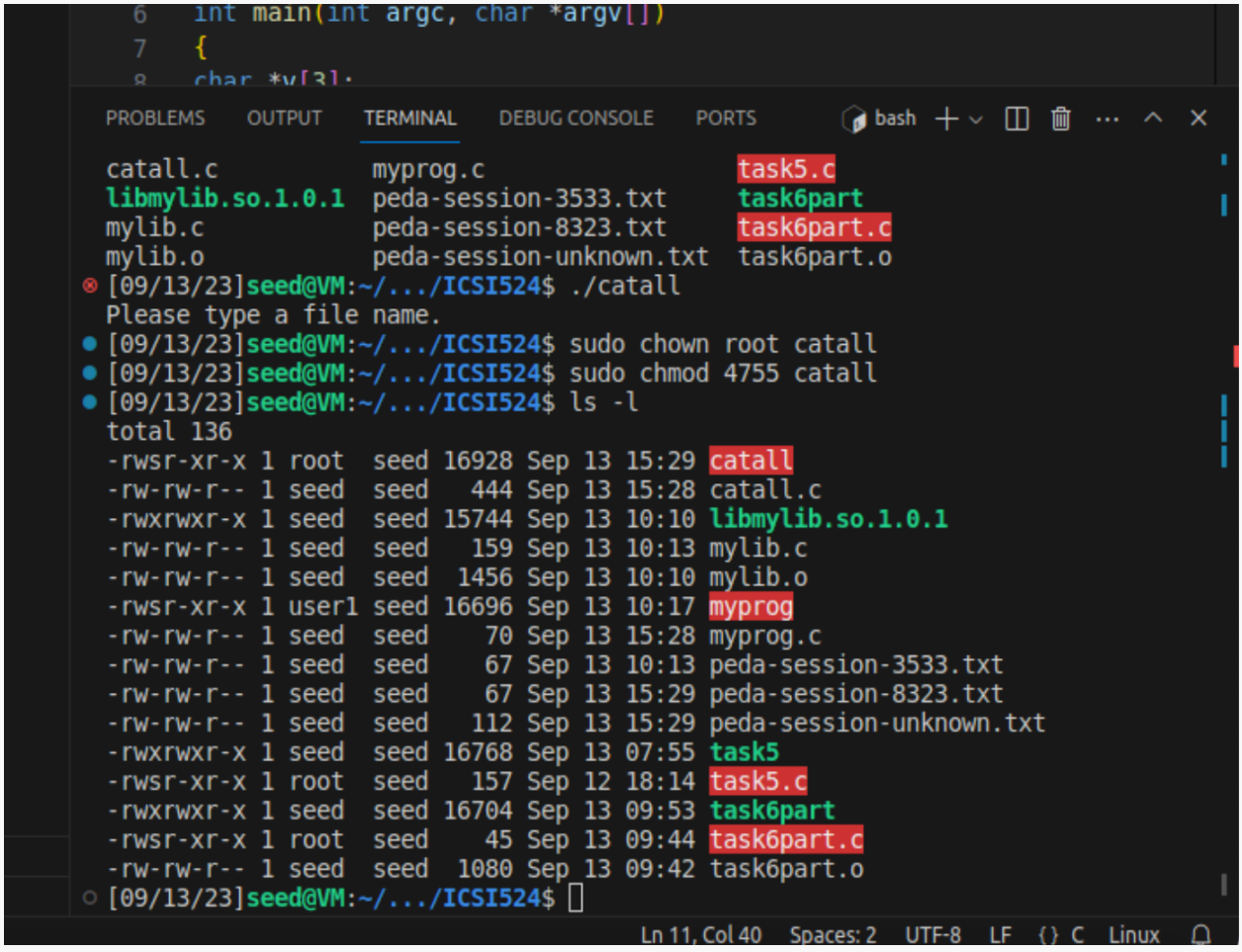
Set UID : sudo chmod 4755 task8

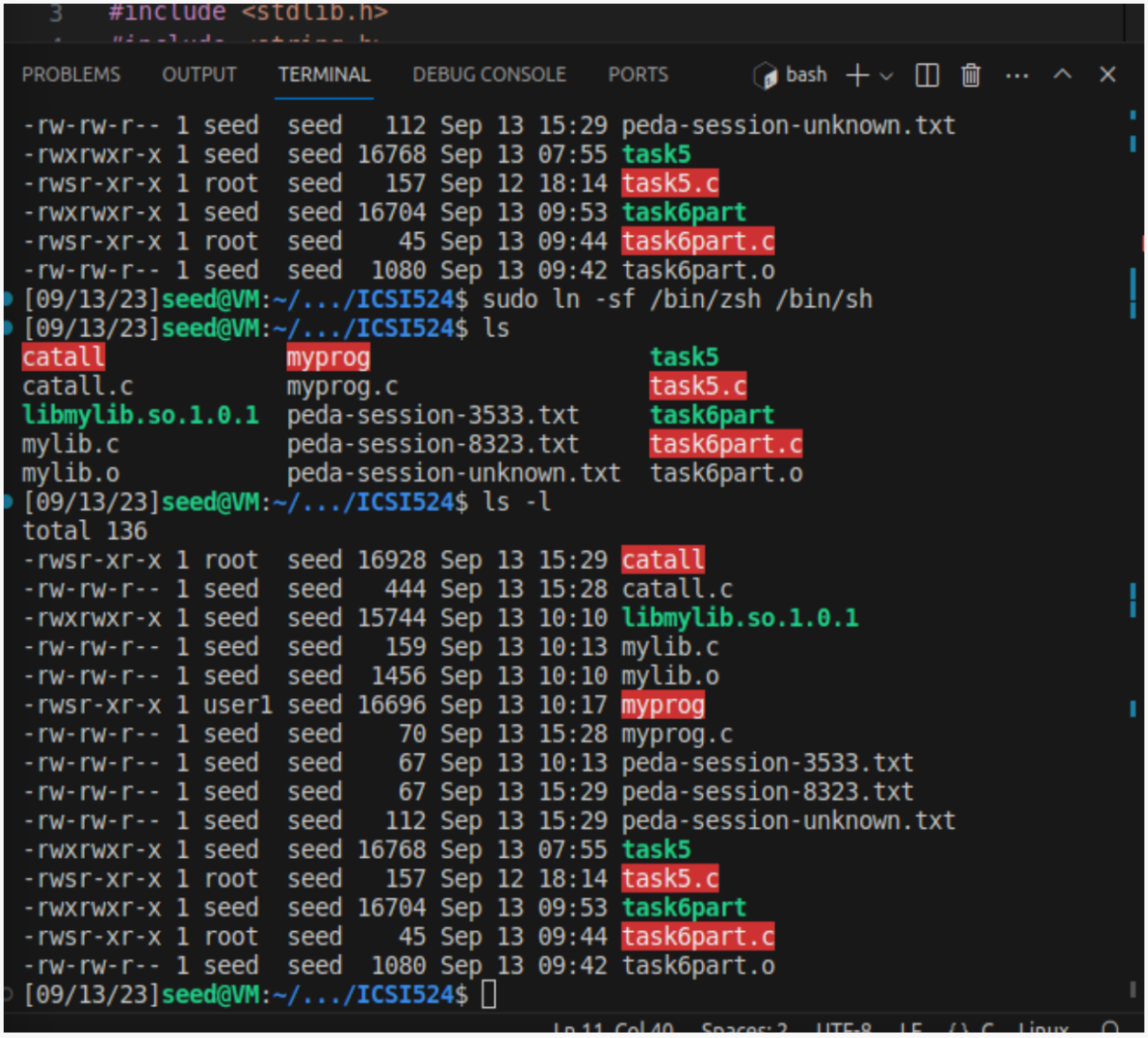
use the following commands to link /bin/sh to /bin/zsh: sudo ln -sf /bin/zsh /bin/sh

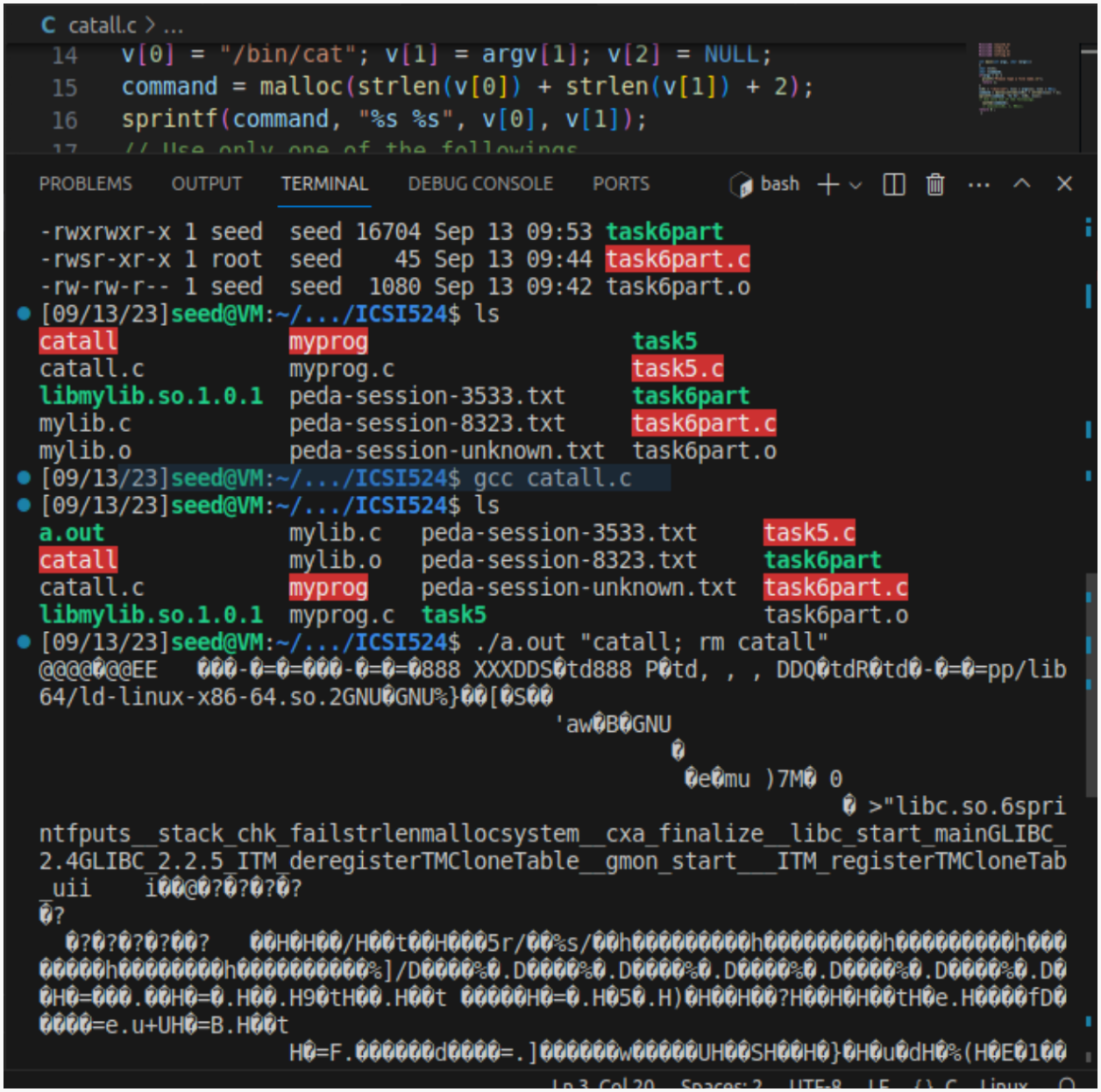
If you were Bob, can you compromise the integrity of the system? For

Can you remove a file that is not writable to you?

Answer : with system() Bob will be able to remove a file. I will prove this by the following screenshot of my result.


6 int main(int argc, char *argv[])
7 {
8 char *v[3];
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS bash
catall.c myprog.c task5.c
libmylib.so.1.0.1 peda-session-3533.txt task6part
mylib.c peda-session-8323.txt task6part.c
mylib.o peda-session-unknown.txt task6part.o
[09/13/23]seed@VM:~/.../ICSI524$ ./catall
Please type a file name.
[09/13/23]seed@VM:~/.../ICSI524$ sudo chown root catall
[09/13/23]seed@VM:~/.../ICSI524$ sudo chmod 4755 catall
[09/13/23]seed@VM:~/.../ICSI524$ ls -l
total 136
-rwsr-xr-x 1 root seed 16928 Sep 13 15:29 catall
-rw-rw-r-- 1 seed seed 444 Sep 13 15:28 catall.c
-rwxrwxr-x 1 seed seed 15744 Sep 13 10:10 libmylib.so.1.0.1
-rw-rw-r-- 1 seed seed 159 Sep 13 10:13 mylib.c
-rw-rw-r-- 1 seed seed 1456 Sep 13 10:10 mylib.o
-rwsr-xr-x 1 user1 seed 16696 Sep 13 10:17 myprog
-rw-rw-r-- 1 seed seed 70 Sep 13 15:28 myprog.c
-rw-rw-r-- 1 seed seed 67 Sep 13 10:13 peda-session-3533.txt
-rw-rw-r-- 1 seed seed 67 Sep 13 15:29 peda-session-8323.txt
-rw-rw-r-- 1 seed seed 112 Sep 13 15:29 peda-session-unknown.txt
-rwxrwxr-x 1 seed seed 16768 Sep 13 07:55 task5
-rwsr-xr-x 1 root seed 157 Sep 12 18:14 task5.c
-rwxrwxr-x 1 seed seed 16704 Sep 13 09:53 task6part
-rwsr-xr-x 1 root seed 45 Sep 13 09:44 task6part.c
-rw-rw-r-- 1 seed seed 1080 Sep 13 09:42 task6part.o
[09/13/23]seed@VM:~/.../ICSI524$
Ln 11, Col 40 Spaces: 2 UTF-8 LF { } C Linux


3 #include <stdlib.h>
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS bash
-rw-rw-r-- 1 seed seed 112 Sep 13 15:29 peda-session-unknown.txt
-rwxrwxr-x 1 seed seed 16768 Sep 13 07:55 task5
-rwsr-xr-x 1 root seed 157 Sep 12 18:14 task5.c
-rwxrwxr-x 1 seed seed 16704 Sep 13 09:53 task6part
-rwsr-xr-x 1 root seed 45 Sep 13 09:44 task6part.c
-rw-rw-r-- 1 seed seed 1080 Sep 13 09:42 task6part.o
[09/13/23]seed@VM:~/.../ICSI524$ sudo ln -sf /bin/zsh /bin/sh
[09/13/23]seed@VM:~/.../ICSI524$ ls
catall myprog task5
catall.c myprog.c task5.c
libmylib.so.1.0.1 peda-session-3533.txt task6part
mylib.c peda-session-8323.txt task6part.c
mylib.o peda-session-unknown.txt task6part.o
[09/13/23]seed@VM:~/.../ICSI524$ ls -l
total 136
-rwsr-xr-x 1 root seed 16928 Sep 13 15:29 catall
-rw-rw-r-- 1 seed seed 444 Sep 13 15:28 catall.c
-rwxrwxr-x 1 seed seed 15744 Sep 13 10:10 libmylib.so.1.0.1
-rw-rw-r-- 1 seed seed 159 Sep 13 10:13 mylib.c
-rw-rw-r-- 1 seed seed 1456 Sep 13 10:10 mylib.o
-rwsr-xr-x 1 user1 seed 16696 Sep 13 10:17 myprog
-rw-rw-r-- 1 seed seed 70 Sep 13 15:28 myprog.c
-rw-rw-r-- 1 seed seed 67 Sep 13 10:13 peda-session-3533.txt
-rw-rw-r-- 1 seed seed 67 Sep 13 15:29 peda-session-8323.txt
-rw-rw-r-- 1 seed seed 112 Sep 13 15:29 peda-session-unknown.txt
-rwxrwxr-x 1 seed seed 16768 Sep 13 07:55 task5
-rwsr-xr-x 1 root seed 157 Sep 12 18:14 task5.c
-rwxrwxr-x 1 seed seed 16704 Sep 13 09:53 task6part
-rwsr-xr-x 1 root seed 45 Sep 13 09:44 task6part.c
-rw-rw-r-- 1 seed seed 1080 Sep 13 09:42 task6part.o
[09/13/23]seed@VM:~/.../ICSI524$



// use gcc catall.c to create a.out

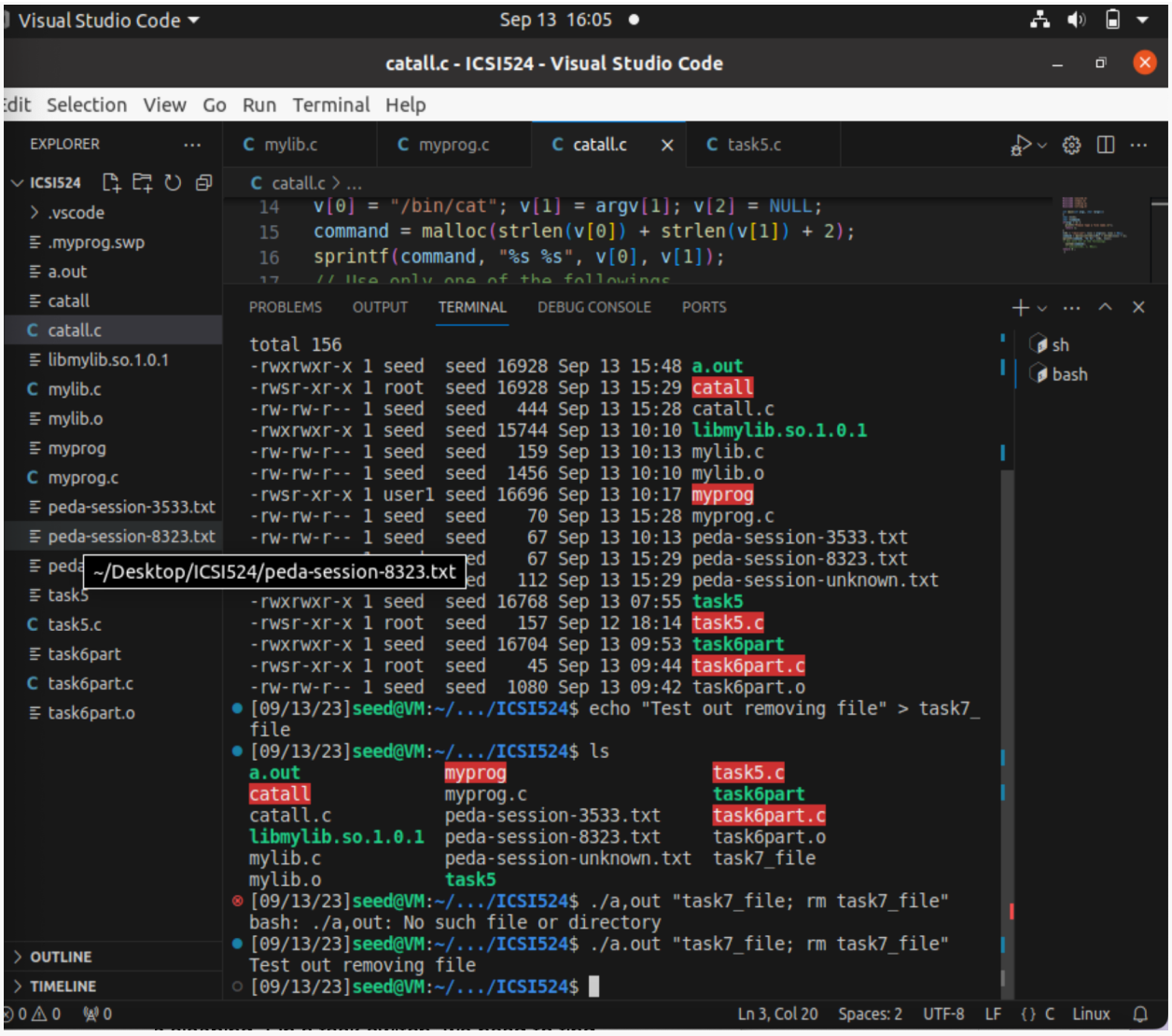
// I created just task7_file by using echo command line

ex) echo “Test out removing file” > test7_file.

// This will create task7_file and I used

./a.out “task7_file; rm task7_file” to remove task7_file

And it actually removed task7_file. (I used ; to put multiple line of command)

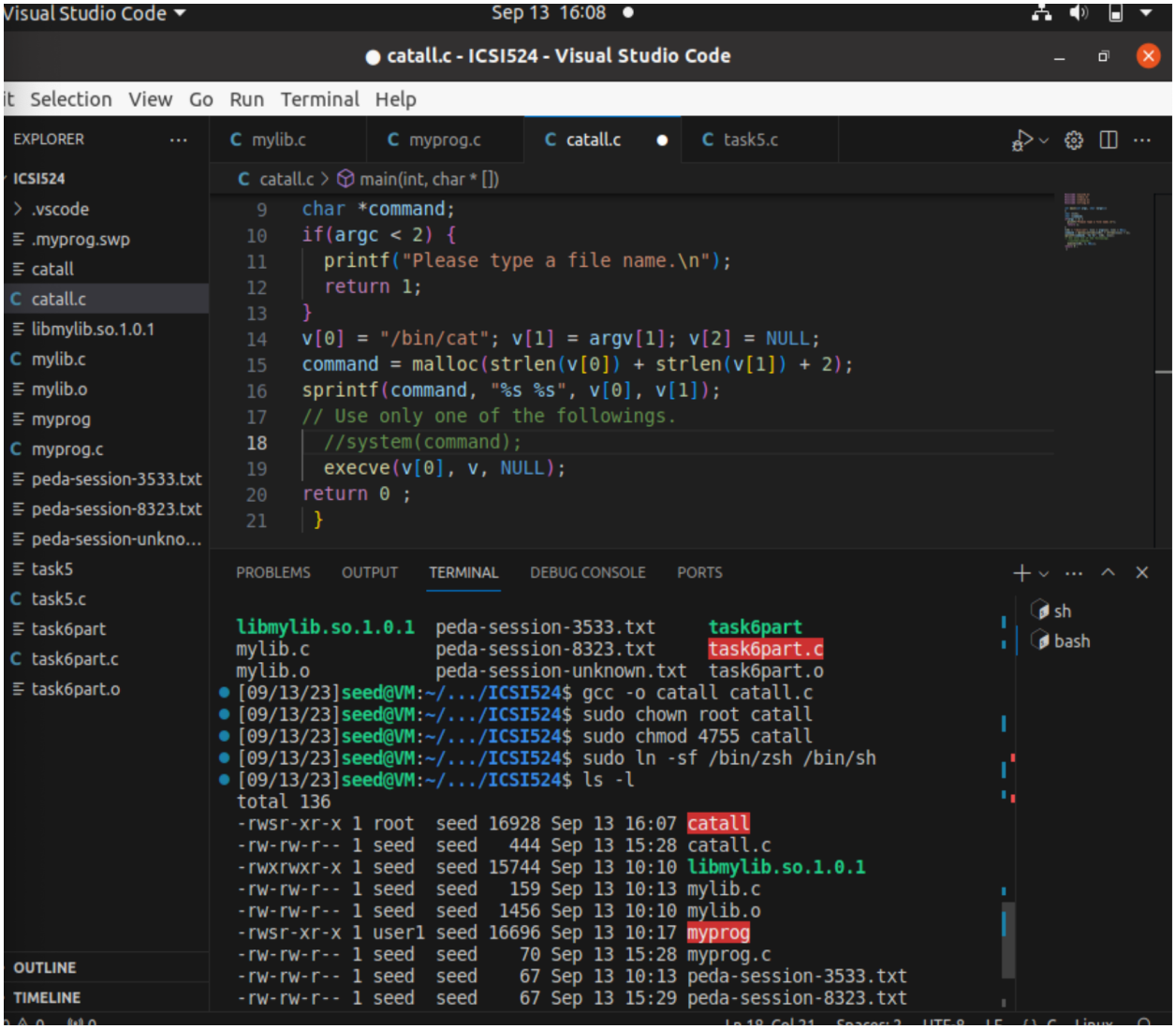


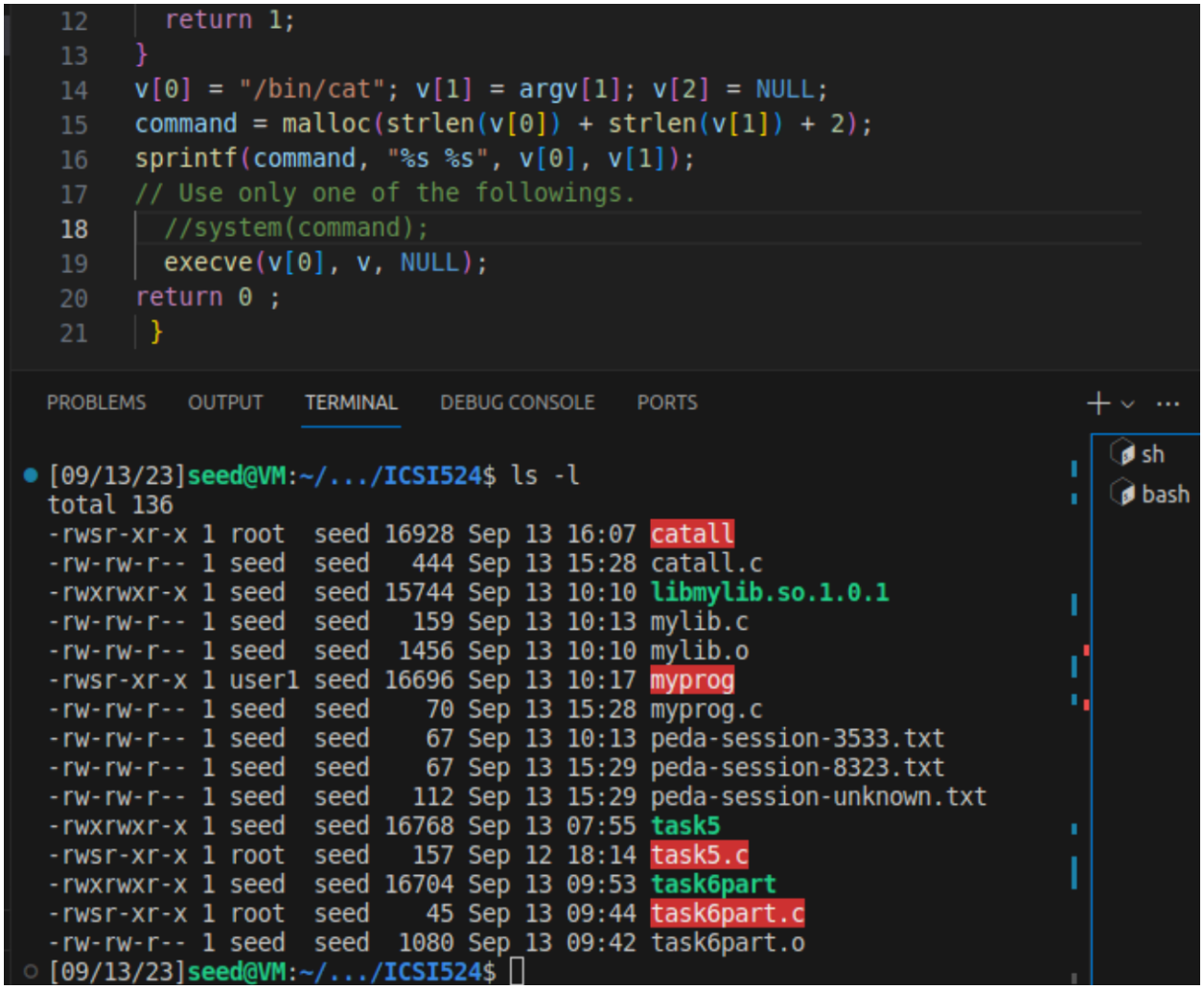
// with system()

Since the system invokes a shell, based on the output that I tried I was able to remove the file, so bob will be able to remove the file.

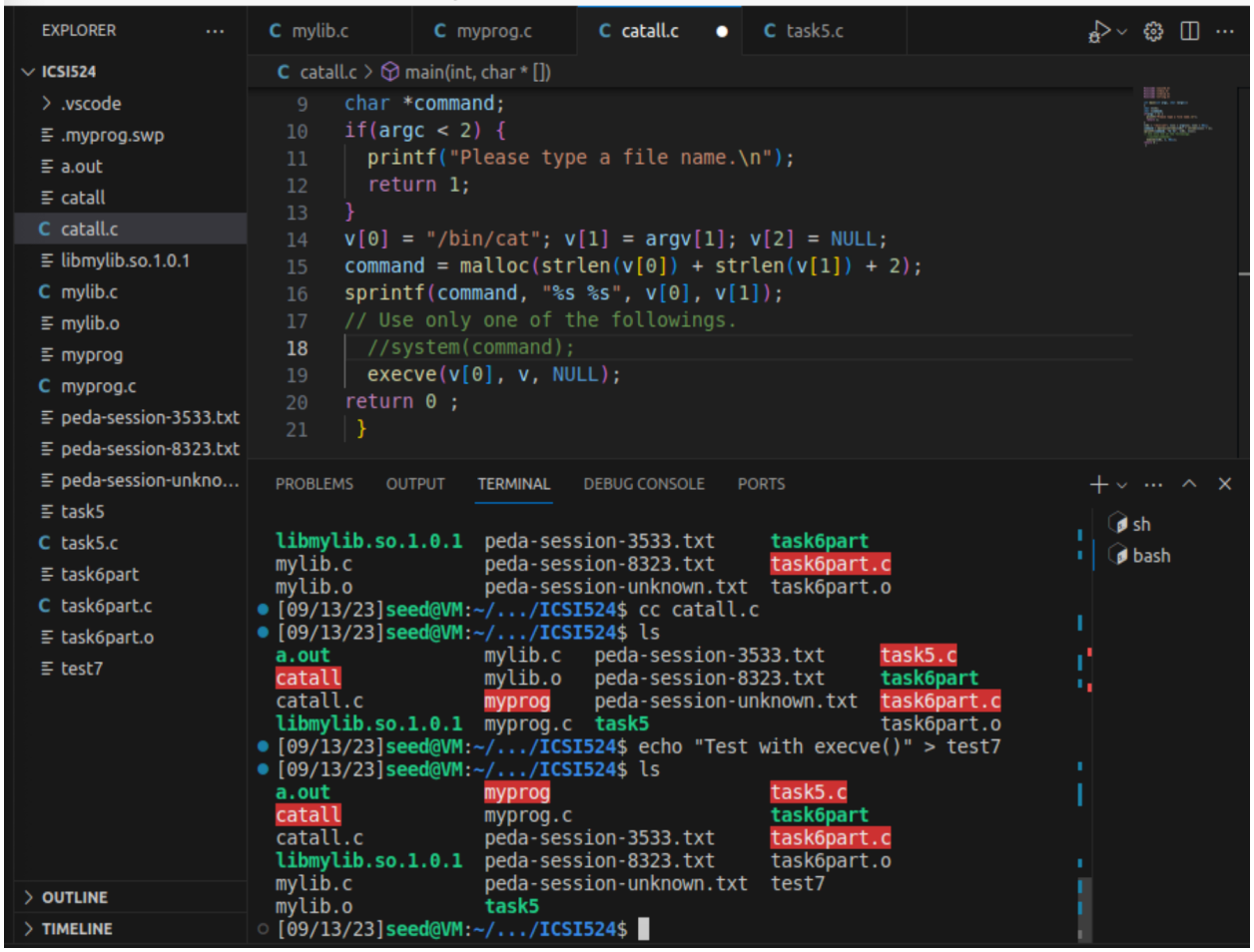
Step 2: Comment out the system(command) statement, and uncomment the execve() statement; the program will use execve() to invoke the command. Compile the program, and make it a root-owned Set-UID. Do your attacks in Step 1 still work? Please describe and explain your observations.

// with execve()


Visual Studio Code
Sep 13 16:08
catall.c - ICSI524 - Visual Studio Code
it Selection View Go Run Terminal Help
EXPLORER
ICSI524
.vscode
.myprog.swp
catall
catall.c
libmylib.so.1.0.1
mylib.c
mylib.o
myprog
myprog.c
peda-session-3533.txt
peda-session-8323.txt
peda-session-unkno...
task5
task5.c
task6part
task6part.c
task6part.o
OUTLINE
TIMELINE
mylib.c
myprog.c
catall.c
task5.c
catall.c > main(int, char * [])
9 char *command;
10 if(argc < 2) {
11 printf("Please type a file name.\n");
12 return 1;
13 }
14 v[0] = "/bin/cat"; v[1] = argv[1]; v[2] = NULL;
15 command = malloc(strlen(v[0]) + strlen(v[1]) + 2);
16 sprintf(command, "%s %s", v[0], v[1]);
17 // Use only one of the followings.
18 //system(command);
19 execve(v[0], v, NULL);
20 return 0 ;
21 }
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
sh
bash
libmylib.so.1.0.1 peda-session-3533.txt task6part
mylib.c peda-session-8323.txt task6part.c
mylib.o peda-session-unknown.txt task6part.o
[09/13/23]seed@VM:~/.../ICSI524$ gcc -o catall catall.c
[09/13/23]seed@VM:~/.../ICSI524$ sudo chown root catall
[09/13/23]seed@VM:~/.../ICSI524$ sudo chmod 4755 catall
[09/13/23]seed@VM:~/.../ICSI524$ sudo ln -sf /bin/zsh /bin/sh
[09/13/23]seed@VM:~/.../ICSI524$ ls -l
total 136
-rwsr-xr-x 1 root seed 16928 Sep 13 16:07 catall
-rw-rw-r-- 1 seed seed 444 Sep 13 15:28 catall.c
-rwxrwxr-x 1 seed seed 15744 Sep 13 10:10 libmylib.so.1.0.1
-rw-rw-r-- 1 seed seed 159 Sep 13 10:13 mylib.c
-rw-rw-r-- 1 seed seed 1456 Sep 13 10:10 mylib.o
-rwsr-xr-x 1 user1 seed 16696 Sep 13 10:17 myprog
-rw-rw-r-- 1 seed seed 70 Sep 13 15:28 myprog.c
-rw-rw-r-- 1 seed seed 67 Sep 13 10:13 peda-session-3533.txt
-rw-rw-r-- 1 seed seed 67 Sep 13 15:29 peda-session-8323.txt

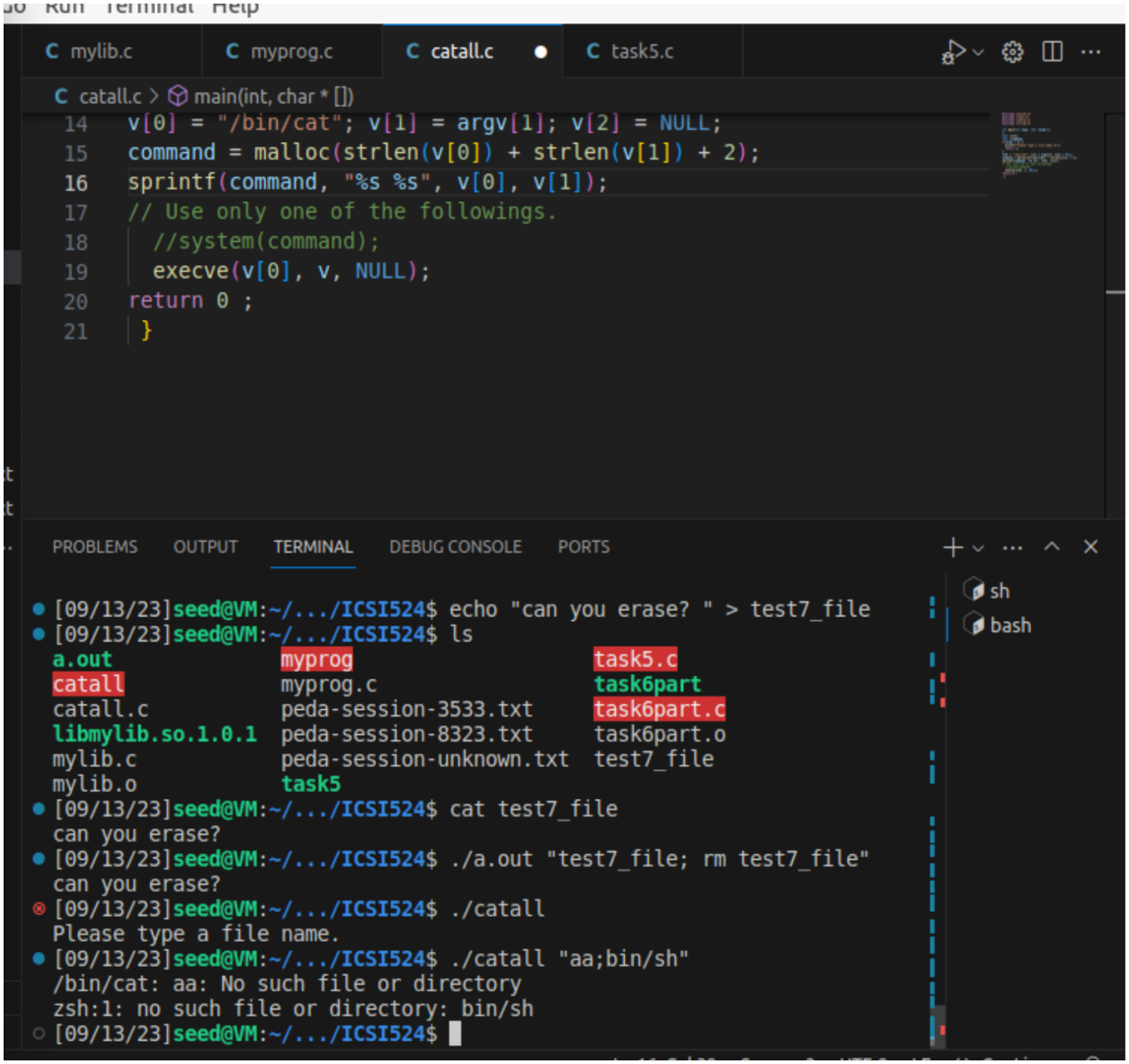


// above I erased a.out and commend out system() and uncomment exeve() and compiled the program again. And then I set the owner to the root and set UID.



// use cc catall.c to create a.out file and created test7 text file using echo command line

// I will try to erase the test7 file.



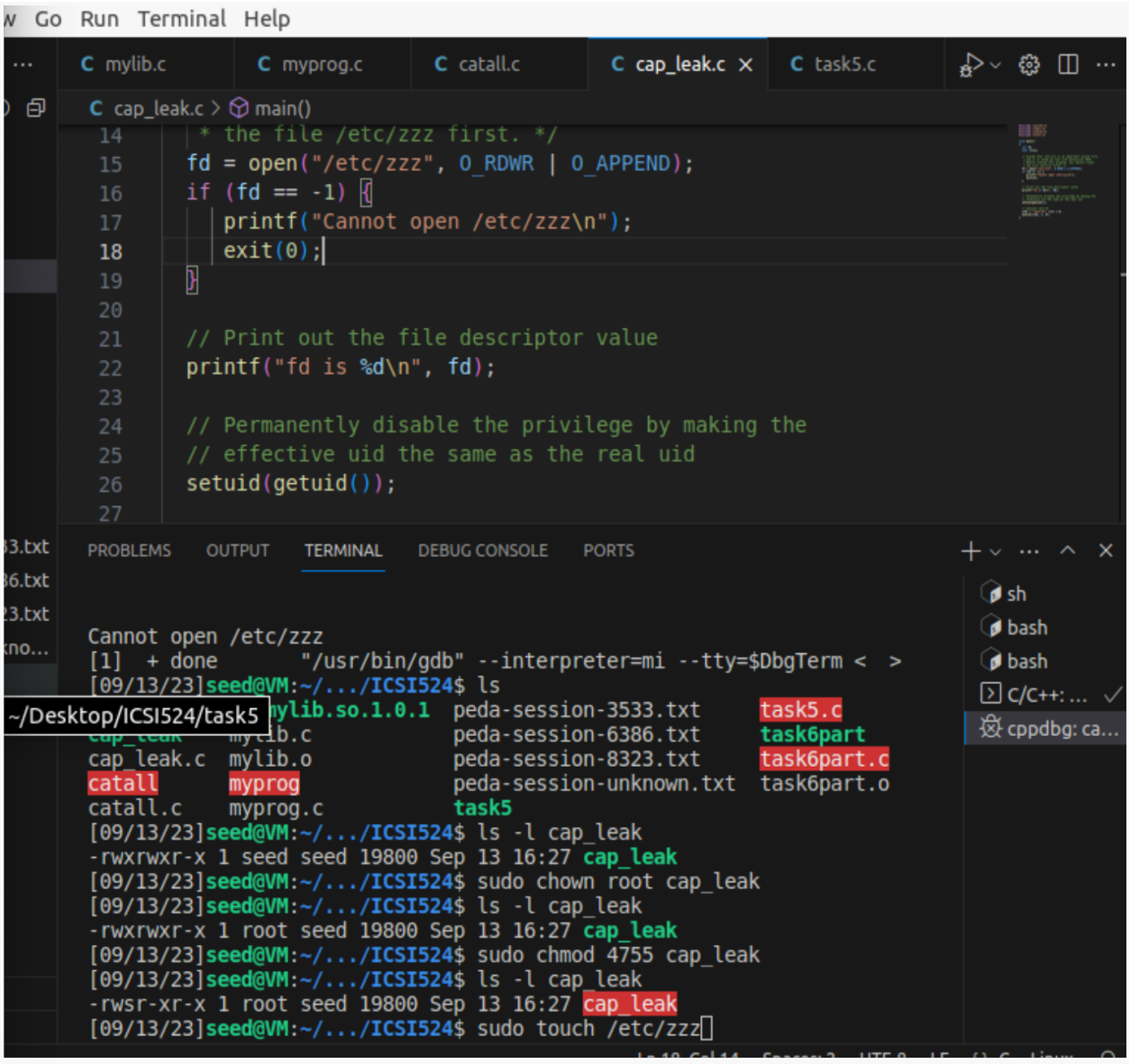
But if I use execve() instead of system(), I wasn't able to remove the file.

When **execve()** is used directly, no shell is invoked so, the accepted grammar is much limited compared to what is accepted by shells. Hence, the entire string is thought of as the path of the file by **cat**. That's why it doesn't work. Therefore this shows that execve() is a better function to use compared to system(), if we think about the security part.

2.9 Task 9: Capability Leaking

Step1: create cap_leak by using gcc cap_leak -o cap_leak. Check permission status and owner and it was seed.

I changed the owner as root (sudo chown root cap_leak) and set UID (sudo chmod 4755 cap_leak) and used ls -l cap_leak to check whether it successfully changed owner and UID. And I am trying to access to /etc/zzz



When using the fork() function, both the child process and the parent process use the same file descriptor. The paranet process closed the fd for "/etc/zzz", but the child process still owns the fd for "/etc/zzz", so the child process can access the fd and write "Malicious Data".