

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           catal1.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 catal1.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          catal1.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.2w8xHH01Ef/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/local/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framework/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/share/dotnet:~/dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/appleinternal/bin:/Users/choeseoyeon/.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

```
/Users/choeseoyeon/Desktop/ICSI524/Lab/lab03-main %
(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.2w8xHH01Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=
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/local/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framework/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/share/dotnet:~/dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/appleinternal/bin:/Users/choeseoyeon/.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.2w8xHH01Ef/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/local/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framework/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/share/dotnet:~/dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptext/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptext/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptext/codex.system/bootstrap/usr/appleinternal/bin:/Users/choeseoyeon/.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.2w8xHH01Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=@
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/local/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framework/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/share/dotnet:~/dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/appleinternal/bin:/Users/choeseoyeon/.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) choeseyeon@dyn-169-226-220-44 lab03-main % gcc myprintenv.c
(base) choeseyeon@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          catal1.c            myenv.c
(base) choeseyeon@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.2w8xHH01Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=@
TERM_PROGRAM=Apple_Terminal
TERM_PROGRAM_VERSION=447
TERM_SESSION_ID=799C7D92-7ECB-4585-BADE-CBBA72606CD9
SHELL=/bin/zsh
HOME=/Users/choeseyeon
LOGNAME=choeseyeon
USER=choeseyeon
PATH=/Users/choeseyeon/anaconda3/bin:/Users/choeseyeon/anaconda3/condabin:/usr/local/bin:/usr/local/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framework/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/share/dotnet:/~/.dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/appleinternal/bin:/Users/choeseyeon/.zsh/misc
SHLVL=1
PWD=/Users/choeseyeon/Desktop/ICSI524/Lab/lab3/lab03-main
OLDPWD=/Users/choeseyeon/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/choeseyeon/anaconda3/bin/conda
_CE_M=
_CE_CONDA=
CONDA_PYTHON_EXE=/Users/choeseyeon/anaconda3/bin/python
CONDA_SHLVL=1
CONDA_PREFIX=/Users/choeseyeon/anaconda3
CONDA_DEFAULT_ENV=base
CONDA_PROMPT_MODIFIER=(base)
LANG=en_US.UTF-8
=/Users/choeseyeon/Desktop/ICSI524/Lab/lab3/lab03-main./a.out
(base) choeseyeon@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           catalle.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.2w8xHH01Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=
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/local/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framework/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/share/dotnet:~/dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/appleinternal/bin:/Users/choeseoyeon/.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.2w8xHH01Ef/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/local/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framework/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/share/dotnet:~/dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptextd/codex.system/bootstrap/usr/appleinternal/bin:/Users/choeseoyeon/.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.

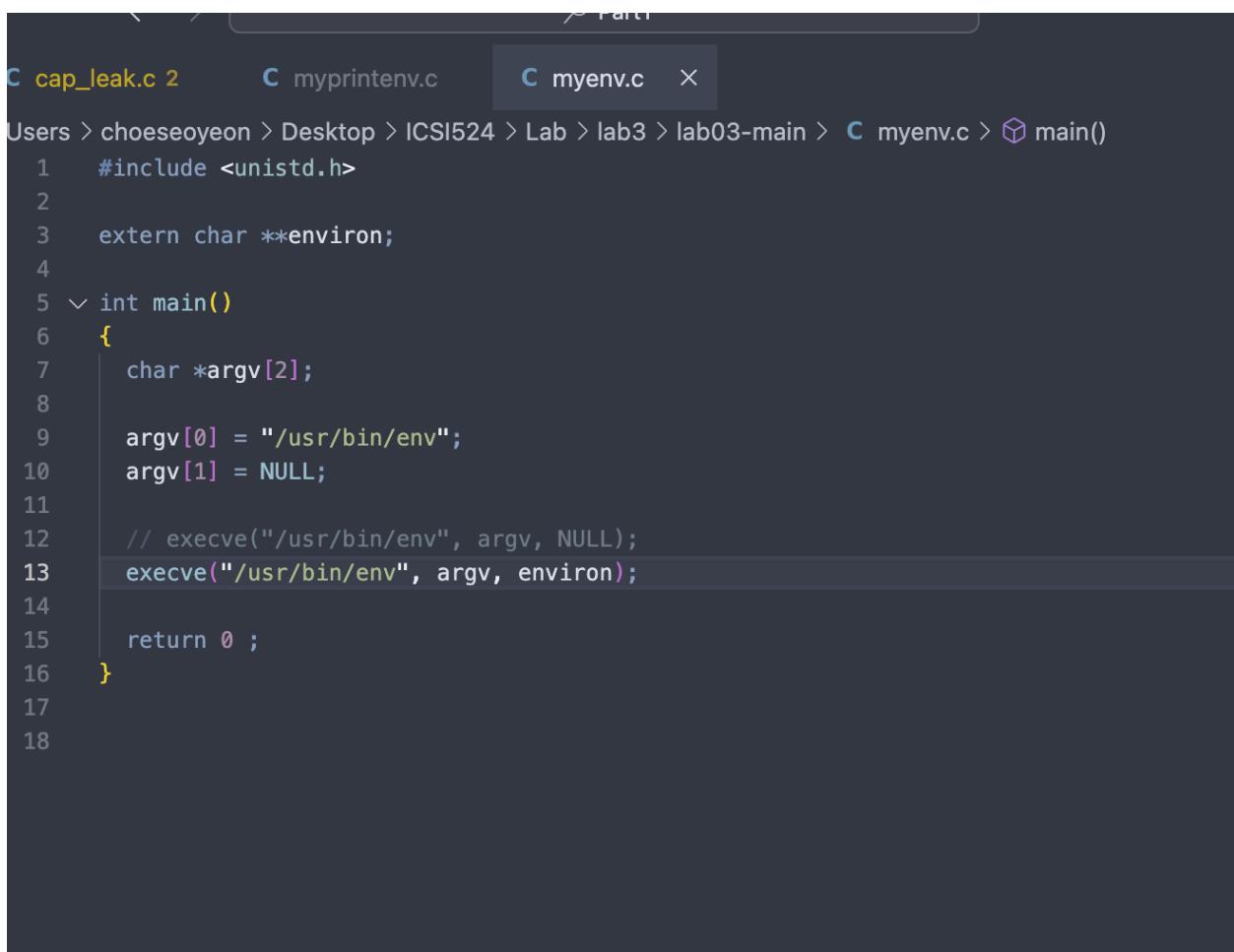
```
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % diff file file2
(base) choeseoyeon@dyn-169-226-220-44 lab03-main % ls
README.md          cap_leak.dSYM        lab03-env-setuid-2.pdf  myprintenv.c
a.out              catall.c           lab03-env-setuid.pdf
cap_leak           file               myenv.c
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.2w8xHH01Ef/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/local/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framework/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/share/dotnet:~/dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptext/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptext/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptext/codex.system/bootstrap/usr/appleinternal/bin:/Users/choeseoyeon/.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.



```
C cap_leak.c 2      C myprintenv.c      C myenv.c ×
Users > choeseoyeon > Desktop > ICSI524 > Lab > lab3 > lab03-main > C myenv.c > main()
1 #include <unistd.h>
2
3 extern char **environ;
4
5 √ int main()
6 {
7     char *argv[2];
8
9     argv[0] = "/usr/bin/env";
10    argv[1] = NULL;
11
12    // execve("/usr/bin/env", argv, NULL);
13    execve("/usr/bin/env", argv, environ);
14
15    return 0 ;
16 }
17
18
```

//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.2w8xHH01Ef/Listeners
SECURITYSESSIONID=18723
XPC_SERVICE_NAME=@
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/local/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framework/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/usr/local/share/dotnet:~/dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptext/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptext/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptext/codex.system/bootstrap/usr/appleinternal/bin:/Users/choeseoyeon/.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.2w8xHH01Ef/Listeners
_CEE_CONDA=
PATH=/Users/choeseoyeon/anaconda3/bin:/Users/choeseoyeon/anaconda3/condabin:/usr/local/bin:/usr/local/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framework/Versions/2.7/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/share/dotnet:~/dotnet/tools:/Library/Apple/usr/bin:/var/run/com.apple.security.cryptext/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptext/codex.system/bootstrap/usr/appleinternal/bin:/Users/choeseoyeon/.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
_CEE_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 exec(), 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.

The screenshot shows the Visual Studio Code interface. The left sidebar has icons for File Explorer, Search, Open, and others. The main editor area displays the code for `task5.c`:

```
home > seed > Desktop > ICSI524 > C task5.c > main()
1 #include <stdio.h>
2 #include <stdlib.h>
3 extern char **environ;
4 int main()
5 {
6     int i = 0;
7     while (environ[i] != NULL) {
8         printf("%s\n", environ[i]);
9     }
10 }
```

The terminal below shows the following command-line session:

- pwd [09/13/23] **seed@VM:~\$** pwd
/home/seed
- [09/13/23] **seed@VM:~\$** cd Desktop
- [09/13/23] **seed@VM:~/Desktop\$** ls
ICSI524
- [09/13/23] **seed@VM:~/Desktop\$** cd ICSI524
- [09/13/23] **seed@VM:~/ICSI524\$** ls
task5.c
- [09/13/23] **seed@VM:~/ICSI524\$** gcc task5.c -o task5
- [09/13/23] **seed@VM:~/ICSI524\$** ls
task5 task5.c
- [09/13/23] **seed@VM:~/ICSI524\$** ./task5

The status bar at the bottom indicates: Ln 10, Col 2, Spaces: 2, UTF-8, LF, { } C, Linux.

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

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** task5.c - Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Left Sidebar:** Includes icons for File Explorer, Search, Find, Open, and others.
- Central Area:** A code editor window showing the following C code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 extern char **environ;
4 int main()
5 {
6     int i = 0;
```
- Terminal Tab:** TERMINAL
- Terminal Output:** Shows the following command-line session:
 - [09/13/23] seed@VM:~/.../ICSI524\$ gcc task5.c -o task5
 - [09/13/23] seed@VM:~/.../ICSI524\$ ls
 - [09/13/23] seed@VM:~/.../ICSI524\$./task5Followed by a large amount of environment variable output:

```
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2038,unix/VM:/tmp/.ICE-unix/2038
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
TERM_PROGRAM_VERSION=1.82.1
XDG_CONFIG_DIRS_VSCODE_SNAP_ORIG=/etc/xdg/xdg-ubuntu:/etc/xdg
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GDK_BACKEND_VSCODE_SNAP_ORIG=
GIO_MODULE_DIR_VSCODE_SNAP_ORIG=
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1998
BAMF_DESKTOP_FILE_HINT=/var/lib/snapd/desktop/applications/code_code.desktop
NO_AT_BRIDGE=1
GTK_MODULES=gail:atk-bridge
PWD=/home/seed/Desktop/ICSI524
GSETTINGS_SCHEMA_DIR=/home/seed/snap/code/139/.local/share/glib-2.0/schemas
XDG_SESSION_DESKTOP=ubuntu
LOGNAME=seed
GTK_EXE_PREFIX=/snap/code/139/usr
```
- Bottom Status Bar:** Ln 10, Col 2 Spaces: 2 UTF-8 LF () C Linux

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

Activities Visual Studio Code Sep 13 07:58

task5.c - Visual Studio Code

File Edit Selection View Go Run Terminal Help

Welcome task5.c

home > seed > Desktop > ICSI524 > C task5.c > main()

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 extern char **environ;
4 int main()
5 {
6     int i = 0;
7 }
```

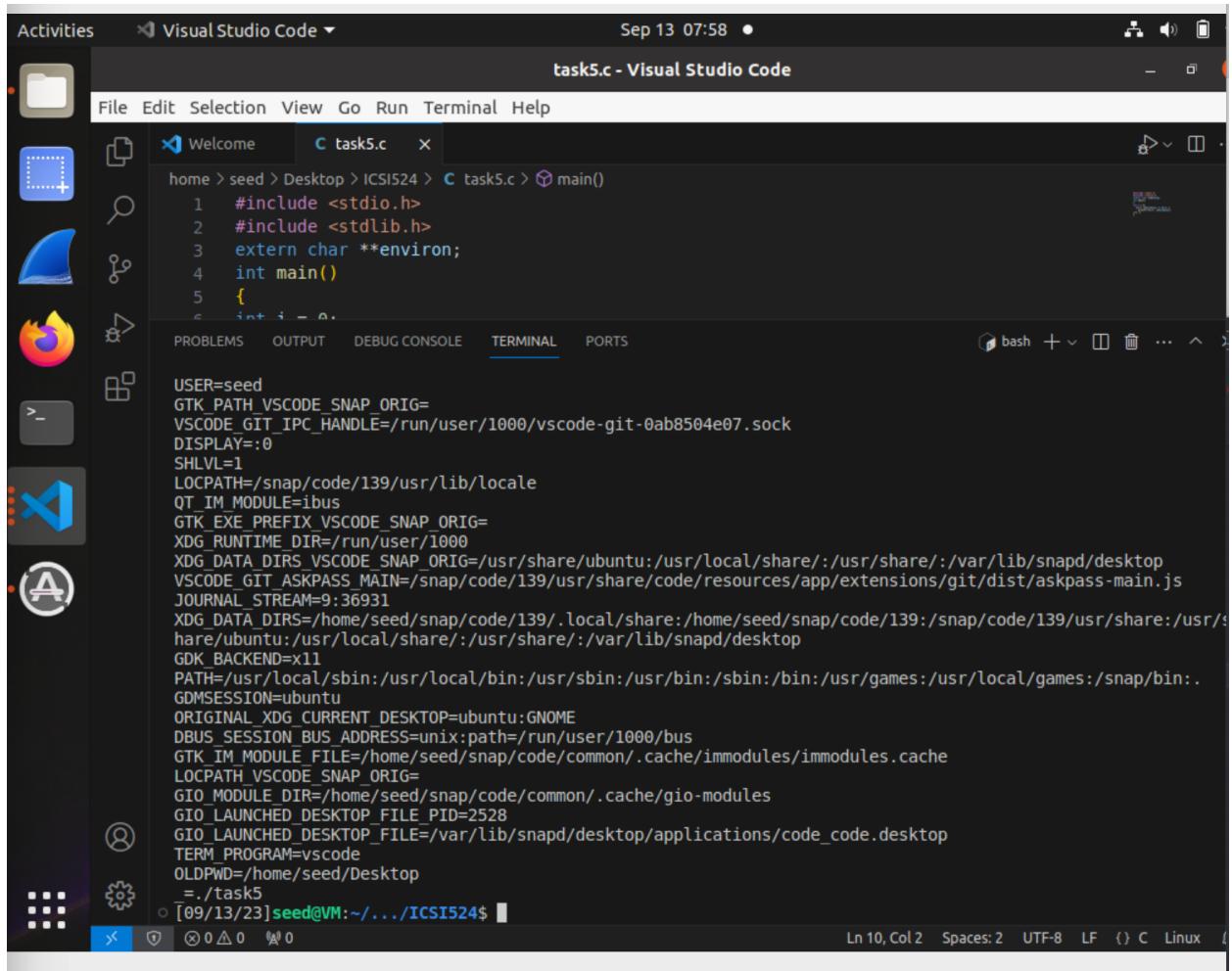
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

bash + ... ^

USER=seed
GTK_PATH_VSCODE_SNAP_ORIG=
VS CODE GIT IPC HANDLE=/run/user/1000/vscode-git-0ab8504e07.sock
DISPLAY=:0
SHLVL=1
LOCPATH=/snap/code/139/usr/lib/locale
QT_IM_MODULE=ibus
GTK_EXE_PREFIX_VSCODE_SNAP_ORIG=
XDG_RUNTIME_DIR=/run/user/1000
XDG_DATA_DIRS_VSCODE_SNAP_ORIG=/usr/share/ubuntu:/usr/local/share:/var/lib/snapd/desktop
VS CODE GIT ASKPASS MAIN=/snap/code/139/usr/share/code/resources/app/extensions/git/dist/askpass-main.js
JOURNAL_STREAM=9:36931
XDG_DATA_DIRS=/home/seed/snap/code/139/.local/share:/home/seed/snap/code/139:/snap/code/139/usr/share:/usr/share/ubuntu:/usr/local/share:/usr/share:/var/lib/snapd/desktop
GDK_BACKEND=x11
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:
GDMSESSION=ubuntu
ORIGINAL_XDG_CURRENT_DESKTOP=ubuntu:GNOME
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
GTK_IM_MODULE_FILE=/home/seed/snap/code/common/.cache/immodules/immodules.cache
LOCPATH_VSCODE_SNAP_ORIG=
GIO_MODULE_DIR=/home/seed/snap/code/common/.cache/gio-modules
GIO_LAUNCHED_DESKTOP_FILE_PID=2528
GIO_LAUNCHED_DESKTOP_FILE=/var/lib/snapd/desktop/applications/code_code.desktop
TERM_PROGRAM=vscode
OLDPWD=/home/seed/Desktop
=./task5

[09/13/23] seed@VM:~/.../ICSI524\$

Ln 10, Col 2 Spaces: 2 LF {} C Linux



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

The screenshot shows a Linux desktop environment with a yellow border around the window. The window title is "task5.c - Visual Studio Code". The main area displays a C code file named "task5.c" with syntax highlighting. Below the code editor is a terminal window showing command-line history. The terminal output includes environment variables like QT_IM_MODULE, GTK_EXE_PREFIX, and XDG_RUNTIME_DIR, followed by a series of "ls" commands showing files in the current directory. The status bar at the bottom right indicates the terminal is in bash mode, with dimensions of 3x23, 2 spaces, UTF-8 encoding, LF line endings, and a Linux system.

```
QT_IM_MODULE=ibus
GTK_EXE_PREFIX_VSCODE_SNAP_ORIG=
XDG_RUNTIME_DIR=/run/user/1000
XDG_DATA_DIRS_VSCODE_SNAP_ORIG=/usr/share/ubuntu:/usr/local/share:/var/lib/snapd/desktop
VSCODE_GIT_ASKPASS_MAIN=/snap/code/139/usr/share/code/resources/app/extensions/git/dist/askpass-main.js
JOURNAL_STREAM=9:36931
XDG_DATA_DIRS=/home/seed/snap/code/139.local/share:/home/seed/snap/code/139:/snap/code/139/usr/share:/usr/hare/ubuntu:/usr/local/share:/usr/share:/var/lib/snapd/desktop
GDK_BACKEND=x11
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:.
GDMSESSION=ubuntu
ORIGINAL_XDG_CURRENT_DESKTOP=ubuntu:GNOME
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
GTK_IM_MODULE_FILE=/home/seed/snap/code/common/.cache/immodules/immodules.cache
LOCPATH_VSCODE_SNAP_ORIG=
GIO_MODULE_DIR=/home/seed/snap/code/common/.cache/gio-modules
GIO_LAUNCHED_DESKTOP_FILE_PID=2528
GIO_LAUNCHED_DESKTOP_FILE=/var/lib/snapd/desktop/applications/code_code.desktop
TERM_PROGRAM=vscode
OLDPWD=/home/seed/Desktop
= ./task5
● [09/13/23]seed@VM:~/.../ICSI524$ ls
task5 task5.c
● [09/13/23]seed@VM:~/.../ICSI524$ ls -l
total 24
-rwxrwxr-x 1 seed seed 16768 Sep 13 07:55 task5
-rw-rw-r-- 1 seed seed 157 Sep 12 18:14 task5.c
○ [09/13/23]seed@VM:~/.../ICSI524$
```

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

The screenshot shows a terminal window in VS Code with the following content:

```
home > seed > Desktop > ICSI524 > C task5.c > environ
1 #include <stdio.h>
2 #include <stdlib.h>
3 extern char **environ;
4 int main()
5 {
6     int i = 0;
GIO_LAUNCHED_DESKTOP_FILE_PID=2528
GIO_LAUNCHED_DESKTOP_FILE=/var/lib/snapd/desktop/applications/code_code.desktop
TERM_PROGRAM=vscode
OLDPWD=/home/seed/Desktop
./task5
● [09/13/23] seed@VM:~/.../ICSI524$ ls
task5 task5.c
● [09/13/23] seed@VM:~/.../ICSI524$ ls -l
total 24
-rwxrwxr-x 1 seed seed 16768 Sep 13 07:55 task5
-rw-rw-r-- 1 seed seed 157 Sep 12 18:14 task5.c
● [09/13/23] seed@VM:~/.../ICSI524$ sudo chown root tasks5.c
chown: cannot access 'tasks5.c': No such file or directory
● [09/13/23] seed@VM:~/.../ICSI524$ sudo chmod 4755 task5.c
● [09/13/23] seed@VM:~/.../ICSI524$ la =l
ls: cannot access '=l': No such file or directory
● [09/13/23] seed@VM:~/.../ICSI524$ ls-l
ls-l: command not found
● [09/13/23] seed@VM:~/.../ICSI524$ ls -l
total 24
-rwxrwxr-x 1 seed seed 16768 Sep 13 07:55 task5
-rw-rw-r-- 1 root seed 157 Sep 12 18:14 task5.c
● [09/13/23] seed@VM:~/.../ICSI524$ sudo chmod 4755 task5.c
● [09/13/23] seed@VM:~/.../ICSI524$ ls -l
total 24
-rwxrwxr-x 1 seed seed 16768 Sep 13 07:55 task5
-rwsr-xr-x 1 root seed 157 Sep 12 18:14 task5.c
○ [09/13/23] seed@VM:~/.../ICSI524$
```

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

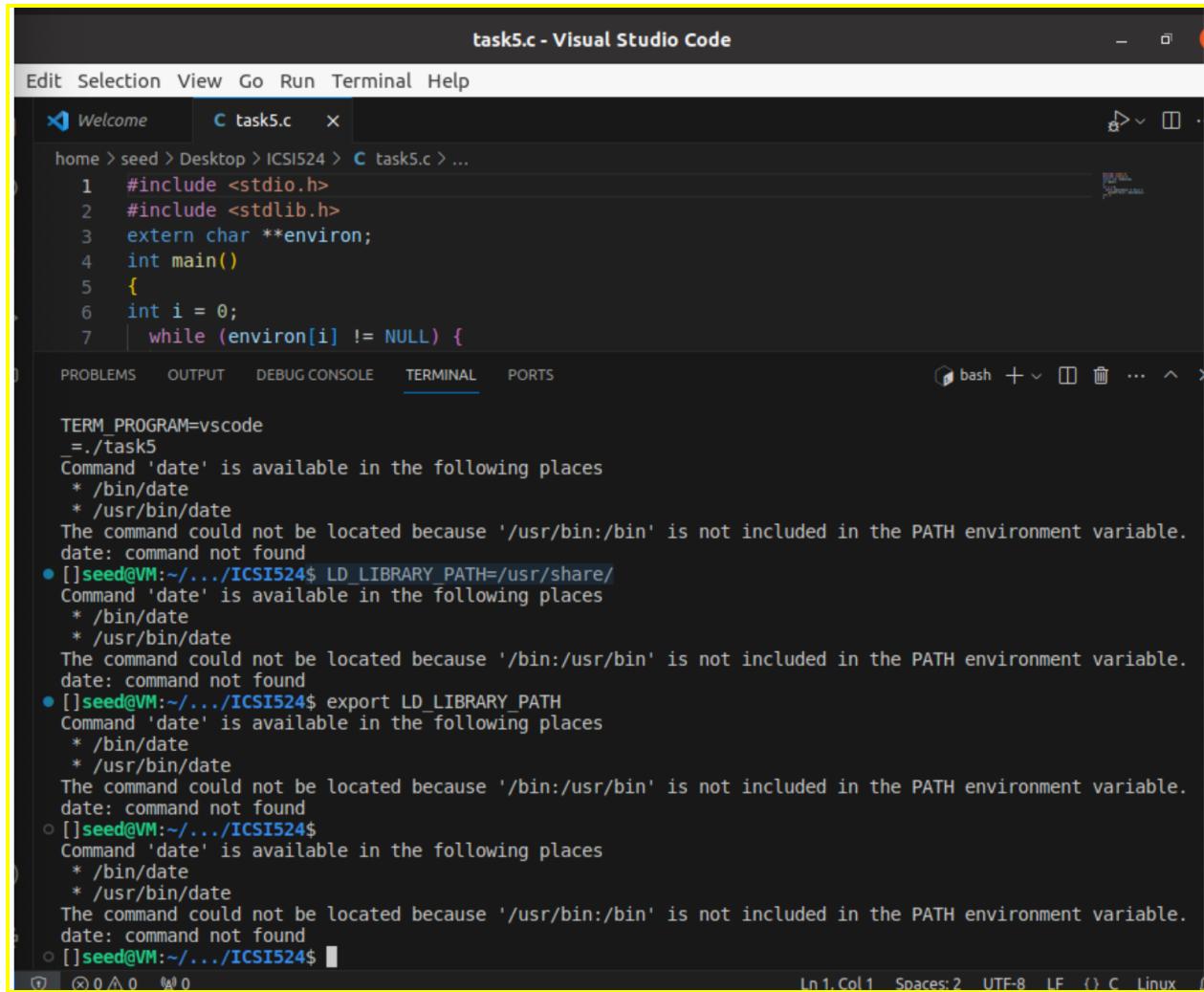
The screenshot shows the Visual Studio Code interface with a yellow border. The title bar reads "task5.c - Visual Studio Code". The menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The left sidebar has icons for Welcome, task5.c, and other files. The main area shows a terminal window with the following output:

```
home > seed > Desktop > ICSI524 > C task5.c > ...
1 #include <stdio.h>
2 #include <stdlib.h>
3 extern char **environ;
4 int main()
5 {
6     int i = 0;
7     while (environ[i] != NULL) {
```

Below the terminal, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS. The status bar at the bottom shows "Ln 1, Col 1 Spaces: 2 UTF-8 LF {} C Linux {".

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

// CHANGE LD_LIBRARY_PATH AND EXPORT



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal tab is selected, showing the command-line history and environment variable changes.

```
task5.c - Visual Studio Code
Edit Selection View Go Run Terminal Help
Welcome task5.c
home > seed > Desktop > ICSI524 > C task5.c > ...
1 #include <stdio.h>
2 #include <stdlib.h>
3 extern char **environ;
4 int main()
5 {
6     int i = 0;
7     while (environ[i] != NULL) {
```

TERMINAL

```
TERM_PROGRAM=vscode
./task5
Command 'date' is available in the following places
 * /bin/date
 * /usr/bin/date
The command could not be located because '/usr/bin:/bin' is not included in the PATH environment variable.
date: command not found
● []seed@VM:~/.../ICSI524$ LD_LIBRARY_PATH=/usr/share/
Command 'date' is available in the following places
 * /bin/date
 * /usr/bin/date
The command could not be located because '/bin:/usr/bin' is not included in the PATH environment variable.
date: command not found
● []seed@VM:~/.../ICSI524$ export LD_LIBRARY_PATH
Command 'date' is available in the following places
 * /bin/date
 * /usr/bin/date
The command could not be located because '/bin:/usr/bin' is not included in the PATH environment variable.
date: command not found
○ []seed@VM:~/.../ICSI524$ 
Command 'date' is available in the following places
 * /bin/date
 * /usr/bin/date
The command could not be located because '/usr/bin:/bin' is not included in the PATH environment variable.
date: command not found
○ []seed@VM:~/.../ICSI524$ 
In 1. Col 1  Spaces:2  UTF-8  LF  {} C  Linux
```

// result

The screenshot shows the Visual Studio Code interface with the title bar "task5.c - Visual Studio Code". The menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help. The top navigation bar has PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The terminal tab displays the following environment variables:

```
XDG_SESSION_CLASS=user
TERM=xterm-256color
GTK_PATH=/snap/code/139/usr/lib/x86_64-linux-gnu/gtk-3.0
LESSOPEN=| /usr/bin/lesspipe %
USER=seed
GTK_PATH_VSCODE_SNAP_ORIG=
VSCODE_GIT_IPC_HANDLE=/run/user/1000/vscode-git-36e1f50e0b.sock
DISPLAY=:0
SHLVL=1
LOCPATH=/snap/code/139/usr/lib/locale
QT_IM_MODULE=ibus
GTK_EXE_PREFIX_VSCODE_SNAP_ORIG=
LD_LIBRARY_PATH=/usr/share/
XDG_RUNTIME_DIR=/run/user/1000
XDG_DATA_DIRS_VSCODE_SNAP_ORIG=/usr/share/ubuntu:/usr/local/share/:/var/lib/snapd/desktop
VSCODE_GIT_ASKPASS_MAIN=/snap/code/139/usr/share/code/resources/app/extensions/git/dist/askpass-main.js
JOURNAL_STREAM=9:36866
XDG_DATA_DIRS=/home/seed/snap/code/139/.local/share:/home/seed/snap/code/139:/snap/code/139/usr/share:/usr/share/ubuntu:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop
GDK_BACKEND=x11
PATH=/usr/local/man/
GDMSESSION=ubuntu
ORIGINAL_XDG_CURRENT_DESKTOP=ubuntu:GNOME
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
GTK_IM_MODULE_FILE=/home/seed/snap/code/common/.cache/immodules/immodules.cache
LOCPATH_VSCODE_SNAP_ORIG=
GIO_MODULE_DIR=/home/seed/snap/code/common/.cache/gio-modules
GIO_LAUNCHED_DESKTOP_FILE_PID=2339
GIO_LAUNCHED_DESKTOP_FILE=/var/lib/snapd/desktop/applications/code_code.desktop
OLDPWD=/home/seed/Desktop
TERM_PROGRAM=vscode
./task5
Command 'date' is available in the following places
 * /bin/date
 * /usr/bin/date
The command could not be located because '/bin:/usr/bin' is not included in the PATH environment variable.
date: command not found
seed@VM:~/.../ICSI524$
```

The status bar at the bottom shows "Ln 2, Col 20" and "Spaces: 2" and "UTF-8".

// 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.

task5.c - Visual Studio Code

File Edit Selection View Go Run Terminal Help

task5.c

home > seed > Desktop > ICSI524 > task5.c > ...

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 extern char **environ;
4 int main()
5 {
6     int i = 0;
7     while (environ[i] != NULL) {
```

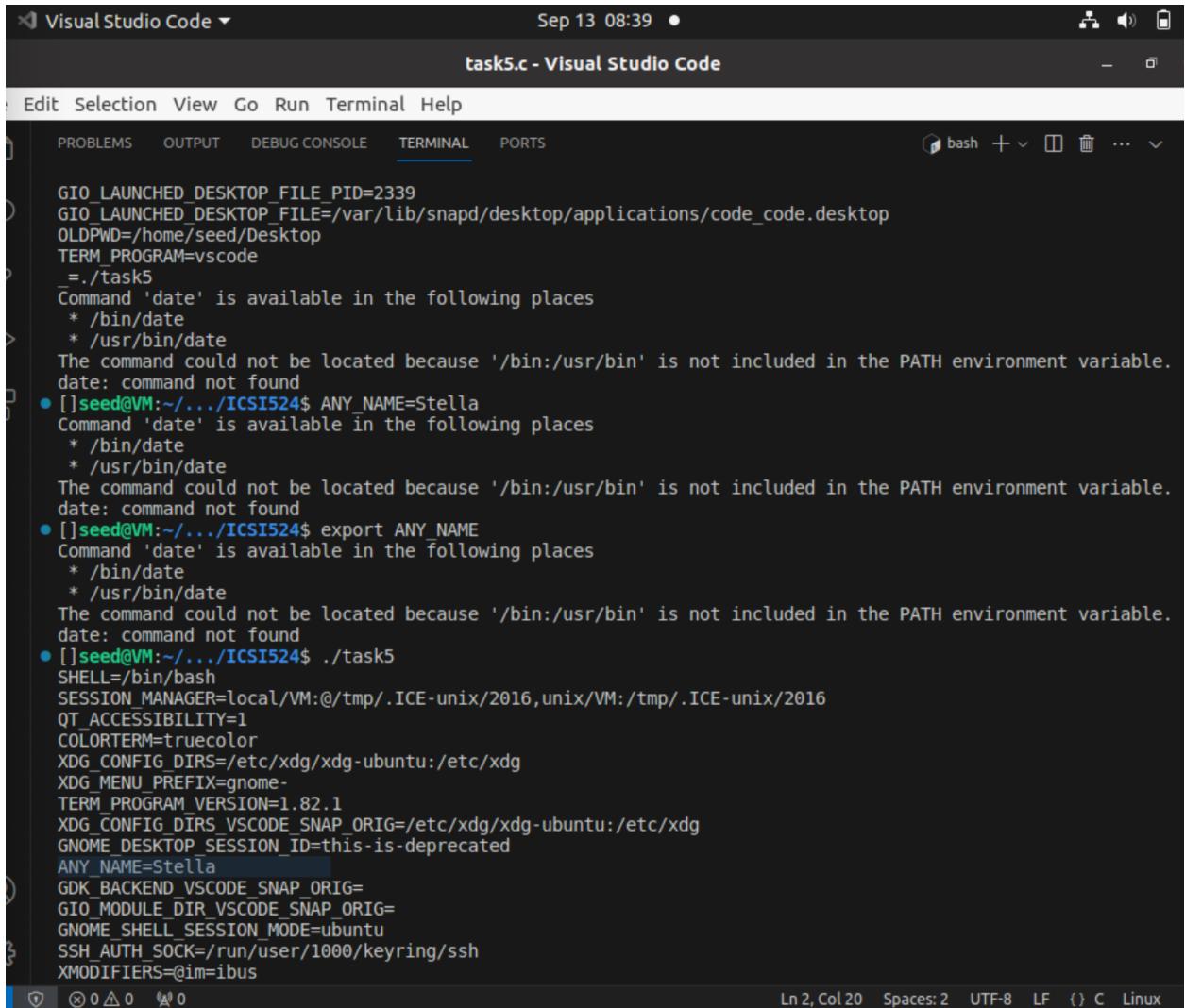
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

date: command not found

- []seed@VM:~/.../ICSI524\$ export LD_LIBRARY_PATH
Command 'date' is available in the following places
 - * /bin/date
 - * /usr/bin/dateThe command could not be located because '/bin:/usr/bin' is not included in the PATH environment variable.
- date: command not found
- []seed@VM:~/.../ICSI524\$
Command 'date' is available in the following places
 - * /bin/date
 - * /usr/bin/dateThe command could not be located because '/usr/bin:/bin' is not included in the PATH environment variable.
- date: command not found
- []seed@VM:~/.../ICSI524\$ ANY_NAME="Stella"
Command 'date' is available in the following places
 - * /bin/date
 - * /usr/bin/dateThe command could not be located because '/usr/bin:/bin' is not included in the PATH environment variable.
- date: command not found
- []seed@VM:~/.../ICSI524\$ export ANY_NAME
Command 'date' is available in the following places
 - * /bin/date
 - * /usr/bin/dateThe command could not be located because '/bin:/usr/bin' is not included in the PATH environment variable.
- date: command not found
- []seed@VM:~/.../ICSI524\$

Ln 2, Col 20 Spaces:2 UTF-8 LF () C Linux

//Result will show that ANY_NAME has been changed.



The screenshot shows a terminal window in Visual Studio Code with the title "task5.c - Visual Studio Code". The terminal tab is selected, and the output shows the following environment variables:

```
GIO_LAUNCHED_DESKTOP_FILE_PID=2339
GIO_LAUNCHED_DESKTOP_FILE=/var/lib/snapd/desktop/applications/code_code.desktop
OLDPWD=/home/seed/Desktop
TERM_PROGRAM=vscode
./task5
Command 'date' is available in the following places
 * /bin/date
 * /usr/bin/date
The command could not be located because '/bin:/usr/bin' is not included in the PATH environment variable.
date: command not found
[]seed@VM:~/.../ICSI524$ ANY_NAME=Stella
Command 'date' is available in the following places
 * /bin/date
 * /usr/bin/date
The command could not be located because '/bin:/usr/bin' is not included in the PATH environment variable.
date: command not found
[]seed@VM:~/.../ICSI524$ export ANY_NAME
Command 'date' is available in the following places
 * /bin/date
 * /usr/bin/date
The command could not be located because '/bin:/usr/bin' is not included in the PATH environment variable.
date: command not found
[]seed@VM:~/.../ICSI524$ ./task5
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2016,unix/VM:/tmp/.ICE-unix/2016
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
TERM_PROGRAM VERSION=1.82.1
XDG_CONFIG_DIRS_VSCODE_SNAP_ORIG=/etc/xdg/ubuntu:/etc/xdg
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
ANY_NAME=Stella
GDK_BACKEND_VSCODE_SNAP_ORIG=
GIO_MODULE_DIR_VSCODE_SNAP_ORIG=
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
```

// 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 .

```
@00*:00)j05y3CE4.00s5888"3      0>0@Z[0
9/20/23] seed@VM:~/.../ICSI524$ ./task6
cap_leak          peda-session-27453.txt
cap_leak.c        peda-session-3533.txt
catall            peda-session-5326.txt
catall.c          peda-session-6386.txt
libmylib.so.1.0.1 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$
```

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

```
C 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    +  ...  ^  X
1>"/tmp/Microsoft-MIEngine-Out-mcmqebgy.roz"
[09/20/23]seed@VM:~/.../ICSI524$ ./ls
your Set-UID program is danger!!
[09/20/23]seed@VM:~/.../ICSI524$ gcc -o task6 task6.c
[09/20/23]seed@VM:~/.../ICSI524$ sudo chown root task6
[09/20/23]seed@VM:~/.../ICSI524$ sudo chmod 4755 task6
[09/20/23]seed@VM:~/.../ICSI524$ task6
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$
```

// 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 C catall.c C cap_ D ...

```
C 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 Linux

The screenshot shows a terminal window with several tabs at the top: 'TERMINAL' (selected), 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'PORTS'. The 'TERMINAL' tab displays a C program named 'ls.c' and its execution output.

```
C 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 }
```

Terminal Output:

```
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$ gcc -o ls ls.c
[09/20/23]seed@VM:~/.../ICSI524$ export PATH=/home/seed:$PATH
[09/20/23]seed@VM:~/.../ICSI524$ echo $PATH
/home/seed:.: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:
/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:...
[09/20/23]seed@VM:~/.../ICSI524$ task6
your Set-UID program is danger!!
[09/20/23]seed@VM:~/.../ICSI524$
```

Bottom status bar: Ln 6, Col 2 Spaces: 4 UTF-8 LF {} C Linu

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

Step1:

The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** mylib.c - ICSI524 - Visual Studio Code
- Menu Bar:** Selection, View, Go, Run, Terminal, Help
- Explorer Bar (Left):** CS1524, libmylib.so.1.0.1, mylib.c, mylib.o, task5, task5.c, task6part, task6part.c, task6part.o
- Editor Area (Top):** Welcome tab selected. The code for `sleep(int)` is displayed:

```
1
2 #include <stdio.h>
3 void sleep (int s)
4 {
5     /* If this is invoked by a privileged program,
6      | you can do damages here! */
7     printf("I am not sleeping!\n");
8 }
```
- Terminal Area (Bottom):** DEBUG CONSOLE tab selected. The terminal output shows the build process:
 - [09/13/23] seed@VM:~/.../ICSI524\$ pwd /home/seed/Desktop/ICSI524
 - [09/13/23] seed@VM:~/.../ICSI524\$ gcc -fPIC -g -c mylib.c
 - [09/13/23] seed@VM:~/.../ICSI524\$ ls mylib.c mylib.o task5 task5.c task6part task6part.c task6part.o
 - [09/13/23] seed@VM:~/.../ICSI524\$ gcc -shared -o libmylib.so.1.0.1 mylib.o -lc
 - [09/13/23] seed@VM:~/.../ICSI524\$ ls libmylib.so.1.0.1 mylib.o task5 task5.c task6part task6part.c mylib.c task5 task6part task6part.o task6part.o
 - [09/13/23] seed@VM:~/.../ICSI524\$ export LD_PRELOAD=./libmylib.so.1.0.1
 - [09/13/23] seed@VM:~/.../ICSI524\$ ls libmylib.so.1.0.1 mylib.o task5 task5.c task6part task6part.c mylib.c task5 task6part task6part.o task6part.o
 - [09/13/23] seed@VM:~/.../ICSI524\$
- Bottom Status Bar:** PROBLEMS, OUTPUT, TERMINAL, DEBUG CONSOLE, PORTS, bash, +, ...

```
// compile myprog.c
```

The screenshot shows a terminal window within a Go workspace interface. The top navigation bar includes 'Go', 'Run', 'Terminal', and 'Help'.

The terminal tab is active, displaying the following C code:

```
C mylib.c > sleep(int)
1 #include <stdio.h>
2 void sleep (int s)
3 {
4     /* If this is invoked by a privileged program,
5      | you can do damages here! */
6     printf("I am not sleeping!\n");
7 }
```

The terminal output pane shows the following build logs:

```
nux-gnu/Scrt1.o: in function `__start':
(.text+0x24): undefined reference to `main'
collect2: error: ld returned 1 exit status
● [09/14/23] seed@VM:~/.../ICSI524$ gcc -fPIC -g -c mylib.c
● [09/14/23] seed@VM:~/.../ICSI524$ gcc -shared -o libmylib.so.1.0
    .1 mylib.o -lc
● [09/14/23] seed@VM:~/.../ICSI524$ ls -l mylib.c
-rw-rw-r-- 1 seed seed 158 Sep 14 13:38 mylib.c
● [09/14/23] seed@VM:~/.../ICSI524$ ls -l
total 208
-rwsr-xr-x 1 root seed 19800 Sep 13 16:27 cap_leak
-rw-rw-r-- 1 seed seed 761 Sep 14 13:26 cap_leak.c
-rwsr-xr-x 1 root seed 16928 Sep 13 16:15 catall
-rw-rw-r-- 1 seed seed 443 Sep 13 16:27 catall.c
-rwxrwxr-x 1 seed seed 19240 Sep 14 13:27 dangerTask6
-rw-rw-r-- 1 seed seed 104 Sep 14 13:26 dangerTask6.c
-rwxrwxr-x 1 seed seed 18696 Sep 14 13:40 libmylib.so.1.0.1
-rw-rw-r-- 1 seed seed 158 Sep 14 13:38 mylib.c
-rw-rw-r-- 1 seed seed 5952 Sep 14 13:40 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 14 13:27 peda-session-26752.t
xt
-rw-rw-r-- 1 seed seed 67 Sep 14 13:32 peda-session-27266.t
xt
-rw-rw-r-- 1 seed seed 67 Sep 14 13:33 peda-session-27453.t
```

The right sidebar contains several icons and labels:

- C/C++: ... (with a gear icon)
- cppdbg: da...
- vim
- bash
- cppdbg: ta...
- bash

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

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

```

● mylib.c - ICSI524 - Visual Studio Code
Edit Selection View Go Run Terminal Help
EXPLORER ... C cap_leak.c C dangerTask6.c C task6part.c ● C mylib.c ...
mylib.c > ...
1 #include <stdio.h>
2 void sleep (int s)
3 {
4     /* If this is invoked by a privileged program,
5        | you can do damages here! */
6     printf("I am not sleeping!\n");
7 }

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
- rwsr-xr-x 1 root seed 157 Sep 12 18:14 task5.c
- rwxrwxr-x 1 seed seed 17760 Sep 14 13:33 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
● [09/14/23]seed@VM:~/.../ICSI524$ export LD_PRELOAD=.libmylib.s
ls-l: command not found
● [09/14/23]seed@VM:~/.../ICSI524$ ls -l
total 208
- rwsr-xr-x 1 root seed 19800 Sep 13 16:27 cap_leak
- rw-rw-r-- 1 seed seed 761 Sep 14 13:26 cap_leak.c
- rwsr-xr-x 1 root seed 16928 Sep 13 16:15 catal
- rw-rw-r-- 1 seed seed 443 Sep 13 16:27 catal.c
- rwxrwxr-x 1 seed seed 19240 Sep 14 13:27 dangerTask6
- rw-rw-r-- 1 seed seed 104 Sep 14 13:26 dangerTask6.c
- rwxrwxr-x 1 seed seed 18696 Sep 14 13:40 libmylib.so.1.0.1
- rw-rw-r-- 1 seed seed 158 Sep 14 13:38 mylib.c
- rw-rw-r-- 1 seed seed 5952 Sep 14 13:40 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 14 13:27 peda-session-26752.t
xt
- rw-rw-r-- 1 seed seed 67 Sep 14 13:32 peda-session-27266.t
xt
- rw-rw-r-- 1 seed seed 67 Sep 14 13:33 peda-session-27453.t

```

// 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"

myprog.c - ICSI524 - Visual Studio Code

dit Selection View Go Run Terminal Help

EXPLORER ... C myprog.c X C catall.c C cap_leak.c C dangerTask6.c C

ICSI524 .vscode .myprog.swp

Source Control (Ctrl+Shift+G) cap_leak

C cap_leak.c catall

C catall.c

C dangerTask6

C dangerTask6.c libmylib.so.1.0.1

C mylib.c mylib.o

E myprog

C myprog.c

E peda-session-3533.txt

E peda-session-6386.txt

E peda-session-8323.txt

E peda-session-26752.txt

E peda-session-27266.txt

E peda-session-27453.txt

E peda-session-unknown.txt

E task5

C task5.c

E task6part

C task6part.c

> OUTLINE

> TIMELINE

C myprog.c > main()

```

1  /* myprog.c */
2  #include <unistd.h>
3  int main()
4  {
5      sleep(1);
6      return 0;
7 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS + v ^

[09/14/23]seed@VM:~/.../ICSI524\$ gcc -fPIC -g -c mylib.c
[09/14/23]seed@VM:~/.../ICSI524\$ export LD_PRELOAD=./libmylib.s
o.1.0.1
[09/14/23]seed@VM:~/.../ICSI524\$./myprog
[09/14/23]seed@VM:~/.../ICSI524\$ sudo chown root myprog
[09/14/23]seed@VM:~/.../ICSI524\$ sudo chmod 4755 myprog
[09/14/23]seed@VM:~/.../ICSI524\$ ls -l myprog
-rwsr-xr-x 1 root seed 16696 Sep 13 10:17 myprog
[09/14/23]seed@VM:~/.../ICSI524\$./myprog
[09/14/23]seed@VM:~/.../ICSI524\$ sudo su
root@VM:/home/seed/Desktop/ICSI524# export LD_PRELOAD=./libmylib
s.o.1.0.1
root@VM:/home/seed/Desktop/ICSI524# ./myprog
I am not sleeping!
root@VM:/home/seed/Desktop/ICSI524#]

bash bash

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"

The screenshot shows a terminal window within a code editor interface. The terminal is running on a VM (VMware) with the command line tool 'seed'. The user has run several commands to set up a user account and change file permissions for a file named 'myprog'. The terminal output is as follows:

```
[09/14/23] seed@VM:~/.../ICSI524$ sudo useradd -d /usr/user1 -m user1
useradd: user 'user1' already exists
● [09/14/23] seed@VM:~/.../ICSI524$ sudo chown user1 myprog
● [09/14/23] seed@VM:~/.../ICSI524$ sudo chmod 4755 myprog
● [09/14/23] seed@VM:~/.../ICSI524$ ls -l myprog
-rwsr-xr-x 1 user1 seed 16696 Sep 13 10:17 myprog
● [09/14/23] seed@VM:~/.../ICSI524$ export LD_PRELOAD=./.libmylib.so.1.0.1
● [09/14/23] seed@VM:~/.../ICSI524$ ./myprog
○ [09/14/23] seed@VM:~/.../ICSI524$
```

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 + ✎ 🗑 ... ⌂ ⌄ X
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
① [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
② [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)

The screenshot shows a Visual Studio Code interface with the title bar "Visual Studio Code" and "Sep 13 16:05". The active tab is "catall.c - ICSI524 - Visual Studio Code". The left sidebar shows a file tree with several files: .vscode, .myprog.swp, a.out, catal, catall.c, libmylib.so.1.0.1, mylib.c, mylib.o, myprog, myprog.c, peda-session-3533.txt, peda-session-8323.txt, peda (~Desktop/ICSI524/peda-session-8323.txt), task5, task5.c, task6part, task6part.c, task6part.o, and task6part.o. The terminal tab is open and displays the following session:

```
total 156
-rwxrwxr-x 1 seed  seed 16928 Sep 13 15:48 a.out
-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$ echo "Test out removing file" > task7_
file
[09/13/23]seed@VM:~/.../ICSI524$ ls
a.out      myprog      task5.c
catall     myprog.c    task6part
catall.c   peda-session-3533.txt  task6part.c
libmylib.so.1.0.1 peda-session-8323.txt  task6part.o
mylib.c    peda-session-unknown.txt  task7_file
mylib.o    task5
[09/13/23]seed@VM:~/.../ICSI524$ ./a.out "task7_file; rm task7_file"
bash: ./a.out: No such file or directory
[09/13/23]seed@VM:~/.../ICSI524$ ./a.out "task7_file; rm task7_file"
Test out removing file
[09/13/23]seed@VM:~/.../ICSI524$
```

```
// 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

File Selection View Go Run Terminal Help

EXPLORER ... C mylib.c C myprog.c C catall.c ● C task5.c

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

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

OUTLINE TIMELINE

```

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

+ ...

- [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
-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\$** []

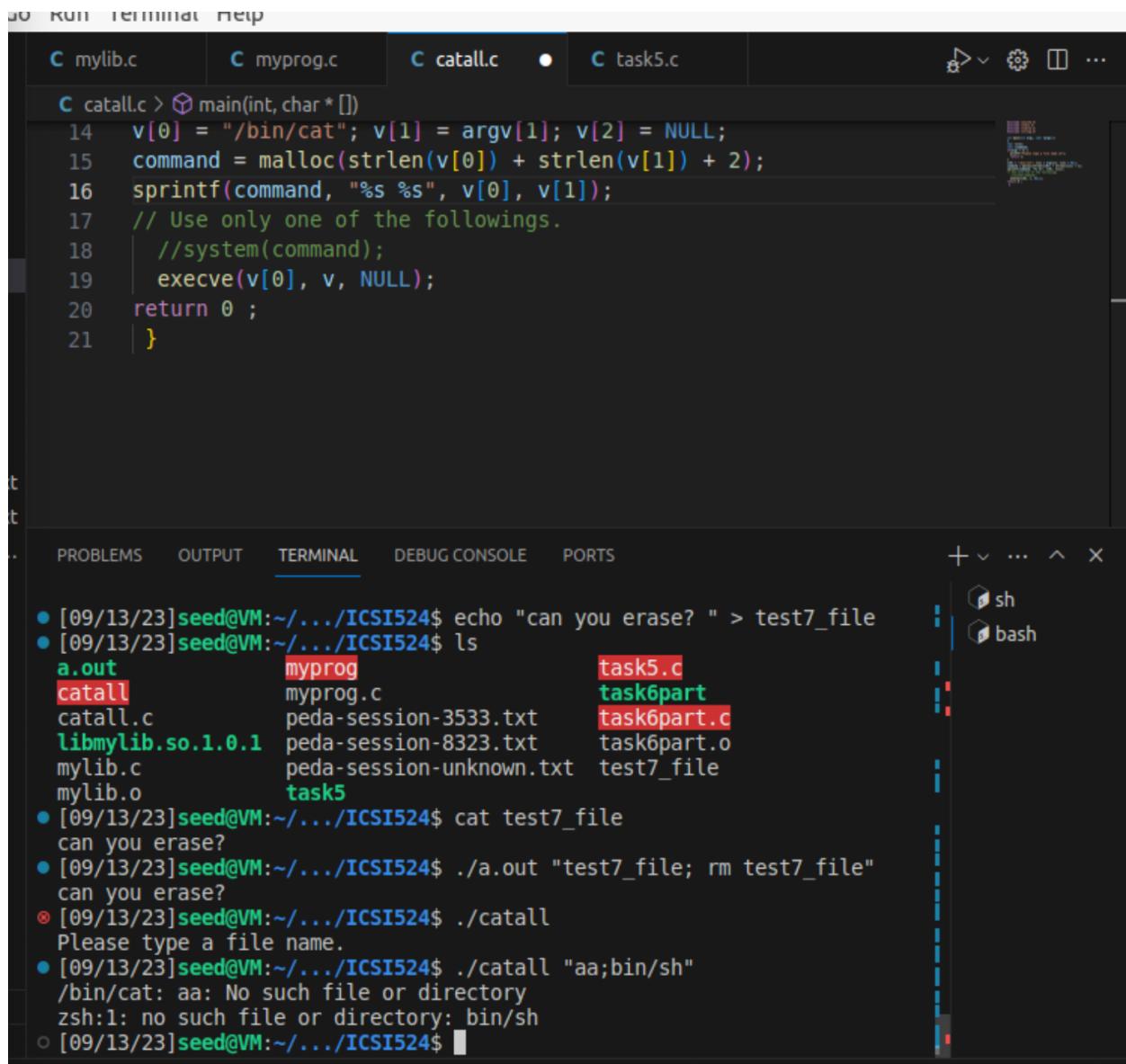
// above I erased a.out and comment out system() and uncomment exeve() and compiled the program again. And then I set the owner to the root and set UID.

The screenshot shows a VS Code interface with the following details:

- EXPLORER** sidebar:
 - Project folder **ICSI524** containing:
 - .vscode
 - .myprog.swp
 - a.out
 - catal
 - C catal.c** (selected)
 - libmylib.so.1.0.1
 - C mylib.c
 - E mylib.o
 - E myprog
 - C myprog.c
 - E peda-session-3533.txt
 - E peda-session-8323.txt
 - E peda-session-unkno...
 - E task5
 - C task5.c
 - E task6part
 - C task6part.c
 - E task6part.o
 - E test7
- CODE** tab bar: mylib.c, myprog.c, catal.c (selected), task5.c.
- TERMINAL** tab bar: PROBLEMS, OUTPUT, TERMINAL (selected), DEBUG CONSOLE, PORTS.
- TERMINAL** content:
 - Session 1: libmylib.so.1.0.1 peda-session-3533.txt task6part
 - Session 2: mylib.c peda-session-8323.txt task6part.c
 - Session 3: mylib.o peda-session-unknown.txt task6part.o
 - Session 4: [09/13/23]seed@VM:~/.../ICSI524\$ cc catal.c
 - Session 5: [09/13/23]seed@VM:~/.../ICSI524\$ ls
 - Session 6: a.out mylib.c peda-session-3533.txt task5.c
 - Session 7: catal mylib.o peda-session-8323.txt task6part
 - Session 8: catal.c myprog peda-session-unknown.txt task6part.c
 - Session 9: libmylib.so.1.0.1 myprog.c task5 task6part.o
 - Session 10: [09/13/23]seed@VM:~/.../ICSI524\$ echo "Test with execve()" > test7
 - Session 11: [09/13/23]seed@VM:~/.../ICSI524\$ ls
 - Session 12: a.out myprog task5.c
 - Session 13: catal myprog.c task6part
 - Session 14: catal.c peda-session-3533.txt task6part.c
 - Session 15: libmylib.so.1.0.1 peda-session-8323.txt task6part.o
 - Session 16: mylib.c peda-session-unknown.txt test7
 - Session 17: mylib.o task5
 - Session 18: [09/13/23]seed@VM:~/.../ICSI524\$
- SIDE BAR**: Shows build artifacts like libmylib.so.1.0.1 and task6part.o.

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

```
// I will try to erase the test7 file.
```



The screenshot shows a terminal window with several tabs at the top: mylib.c, myprog.c, catall.c (which is currently selected), and task5.c. Below the tabs is a code editor window displaying the content of catall.c. The code implements a command-line argument parser and uses `system` to execute the command. The terminal below shows a series of commands being run, including `echo`, `ls`, and `cat` to verify the functionality. A file browser sidebar on the right lists files like a.out, myprog, task5.c, etc.

```

C catall.c > main(int, char *[])
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 + v ... ^ x
● [09/13/23]seed@VM:~/.../ICSI524$ echo "can you erase? " > test7_file
● [09/13/23]seed@VM:~/.../ICSI524$ ls
a.out          myprog           task5.c
catall         myprog.c        task6part
catall.c       peda-session-3533.txt task6part.c
libmylib.so.1.0.1 peda-session-8323.txt task6part.o
mylib.c        peda-session-unknown.txt test7_file
mylib.o        task5

● [09/13/23]seed@VM:~/.../ICSI524$ cat test7_file
can you erase?
● [09/13/23]seed@VM:~/.../ICSI524$ ./a.out "test7_file; rm test7_file"
can you erase?
● [09/13/23]seed@VM:~/.../ICSI524$ ./catall
Please type a file name.
● [09/13/23]seed@VM:~/.../ICSI524$ ./catall "aa;bin/sh"
/bin/cat: aa: No such file or directory
zsh:1: no such file or directory: bin/sh
○ [09/13/23]seed@VM:~/.../ICSI524$ 

```

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

The screenshot shows a terminal window with the following content:

```

mylib.c      myprog.c      catal.c      cap_leak.c      task5.c
...          ...          ...          ...          ...
C cap_leak.c > main()
14     * the file /etc/zzz first. */
15     fd = open("/etc/zzz", O_RDWR | O_APPEND);
16     if (fd == -1) {
17         printf("Cannot open /etc/zzz\n");
18         exit(0);
19     }
20
21     // Print out the file descriptor value
22     printf("fd is %d\n", fd);
23
24     // Permanently disable the privilege by making the
25     // effective uid the same as the real uid
26     setuid(getuid());
27

PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE    PORTS
+ - ... ^ x
sh
bash
bash
C/C++: ... ✓
cppdbg: ca...

```

The terminal output shows:

```

Cannot open /etc/zzz
[1] + done      "/usr/bin/gdb" --interpreter=mi --tty=$DbgTerm < >
[09/13/23]seed@VM:~/.../ICSI524$ ls
~/Desktop/ICSI524/task5 mylib.so.1.0.1 peda-session-3533.txt      task5.c
cap_leak.c   mylib.o      peda-session-6386.txt      task6part
catal.c      myprog       peda-session-8323.txt      task6part.c
catal.c      myprog.c     peda-session-unknown.txt  task6part.o
task5

[09/13/23]seed@VM:~/.../ICSI524$ ls -l cap_leak
-rwxrwxr-x 1 seed seed 19800 Sep 13 16:27 cap_leak
[09/13/23]seed@VM:~/.../ICSI524$ sudo chown root cap_leak
[09/13/23]seed@VM:~/.../ICSI524$ ls -l cap_leak
-rwxrwxr-x 1 root seed 19800 Sep 13 16:27 cap_leak
[09/13/23]seed@VM:~/.../ICSI524$ sudo chmod 4755 cap_leak
[09/13/23]seed@VM:~/.../ICSI524$ ls -l cap_leak
-rwsr-xr-x 1 root seed 19800 Sep 13 16:27 cap_leak
[09/13/23]seed@VM:~/.../ICSI524$ sudo touch /etc/zzz

```

When using the fork() function, both the child process and the parent process use the same file descriptor. The parent 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".