

Task 1: Manipulating Environment Variables

For this task I used `printenv` and `env` commands and they display very similar things. Then I played around with the way the `export` and `unset` commands work as well as using `grep` to display a single variable.

```

root@VM: /etc
Paragraphs
Important stuff
Don't read this stuff pleaseMalicious Data
[02/27/19]seed@VM:~/.../Lab 3$ printenv
XDG_VTNR=7
XDG_SESSION_ID=c1
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
ANDROID_HOME=/home/seed/android/android-sdk-linux
GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1
TERM=xterm-256color
VTE_VERSION=4205
SHELL=/bin/bash
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
LD_PRELOAD=/libmylib.so.1.0.1
WINDOWID=6501722
OLDPWD=/home/seed/Documents
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1157
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=seed
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:01:cd=40;33:01:or=40;31:01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lzh=01;31:*.lzm=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.diz=01;31:*.gz=01;31:*.lrz=01;31:*.lzo=01;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.taz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.m4=01;35:*.m4v=01;35:*.mp4=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
QT_ACCESSIBILITY=1
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/usr/share/upstart/xdg:/etc/xdg
DESKTOP_SESSION=ubuntu
PATH=/home/seed:/home/seed:/home/seed:/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin:/home/seed/android/android-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-tools:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
QT_IM_MODULE=ibus
QT_OPA_PLATFORMTHEME=appmenu-qt5
task5=task5 env variable
XDG_SESSION_TYPE=x11
PWD=/home/seed/Documents/Lab 3
JOB=unity-settings-daemon
XMODIFIERS=@im=ibus
JAVA_HOME=/usr/lib/jvm/java-8-oracle
GNOME_KEYRING_PID=

```

```

root@VM: /etc
QT_IM_MODULE=ibus
QT_OPA_PLATFORMTHEME=appmenu-qt5
task5=task5 env variable
XDG_SESSION_TYPE=x11
PWD=/home/seed/Documents/Lab 3
JOB=unity-settings-daemon
XMODIFIERS=@im=ibus
JAVA_HOME=/usr/lib/jvm/java-8-oracle
GNOME_KEYRING_PID=
LANG=en_US.UTF-8
GDM_LANG=en_US
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
COMPIZ_CONFIG_PROFILE=ubuntu
IM_CONFIG_PHASE=1
GDMSESSION=ubuntu
SESSIONTYPE=gnome-session
GTK2_MODULES=overlay-scrollbar
SHLVL=1
HOME=/home/seed
XDG_SEAT=seat0
LANGUAGE=en_US
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
UPSTART_INSTANCE=
UPSTART_EVENTS=xsession started
XDG_SESSION_DESKTOP=ubuntu
LOGNAME=seed
COMPIZ_BIN_PATH=/usr/bin/
dbus SESSION BUS ADDRESS=unix:abstract=/tmp/dbus-H0q7s4p2b6
J2SDKDIR=/usr/lib/jvm/java-8-oracle
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share:/usr/share:/var/lib/snapd/desktop
QT4_IM_MODULE=xim
LESSOPEN=| /usr/bin/lesspipe %s
INSTANCE=
UPSTART_JOB=unity7
XDG_RUNTIME_DIR=/run/user/1000
DISPLAY=:0
XDG_CURRENT_DESKTOP=Unity
GTK_IM_MODULE=ibus
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
LESSCLOSE=/usr/bin/lesspipe %s %s
XAUTHORITY=/home/seed/.Xauthority
=/usr/bin/env
[02/27/19]seed@VM:~/.../Lab 3$ printenv PWD
/home/seed/Documents/Lab 3
[02/27/19]seed@VM:~/.../Lab 3$ env | grep PWD
PWD=/home/seed/Documents/Lab 3
[02/27/19]seed@VM:~/.../Lab 3$ CUSTOMNAME='Test Environment Variable'
[02/27/19]seed@VM:~/.../Lab 3$ echo $CUSTOMNAME
Test Environment Variable
[02/27/19]seed@VM:~/.../Lab 3$ unset CUSTOMNAME
[02/27/19]seed@VM:~/.../Lab 3$ echo $CUSTOMNAME
[02/27/19]seed@VM:~/.../Lab 3$ export NEWCUSTOMNAME='Export Command'
[02/27/19]seed@VM:~/.../Lab 3$ echo $NEWCUSTOMNAME
Export Command
[02/27/19]seed@VM:~/.../Lab 3$ NEWCUSTOMNAME=
[02/27/19]seed@VM:~/.../Lab 3$ echo $NEWCUSTOMNAME
[02/27/19]seed@VM:~/.../Lab 3$

```

Task2: Passing Environment Variables from Parent Process to Child Process

Step 1: For this task, I was to compile and run the following program and describe my observation. Because the output contains many strings, I saved the output into a child file.

Program: task2.c

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
extern char **environ;

void printenv()
{
    int i = 0;
    while (environ[i] != NULL) {
        printf("%s\n", environ[i]);
        i++;
    }
}

void main()
{
    pid_t childPid;

    switch(childPid = fork()) {
        case 0: /*child process*/
            //printenv();
            exit(0);
        default: /*parent process*/
            printenv();
            exit(0);
    }
}
```

```
[02/26/19]seed@VM:~/.../Lab 3$ gcc -o task2 task2.c
[02/26/19]seed@VM:~/.../Lab 3$ task2 > child
[02/26/19]seed@VM:~/.../Lab 3$ ./task2
XDG_VTNR=7
XDG_SESSION_ID=c1
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
ANDROID_HOME=/home/seed/android/android-sdk-linux
GPG_AGENT_INFO=/home/seed/.gnupg/s.gpg-agent:0:1
TERM=xterm-256color
VTE_VERSION=4205
SHELL=/bin/bash
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libboost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0
WINDOWID=65011722
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1157
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=seed
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:01:cd=40;33:01:or=40;31:01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.t7z=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
QT_ACCESSIBILITY=1
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/usr/share/upstart/xdg:/etc/xdg
DESKTOP_SESSION=ubuntu
PATH=/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin:/home/seed/android/android-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-tools:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
QT_IM_MODULE=ibus
QT_QPA_PLATFORMTHEME=appmenu-qt5
```

Observation: The program printed out the environment variables. It saved this output to parent.

Step 2: Now I comment out the `printenv()` statement in the child process case, and uncomment the `printenv()` statement in the parent process case. I then compiled and ran the code again, but this time I changed the output to parent file.

```
[02/26/19]seed@VM:~/.../Lab 3$ gcc -o task2 task2.c
[02/26/19]seed@VM:~/.../Lab 3$ task2 > parent
[02/26/19]seed@VM:~/.../Lab 3$ ./task2
XDG_VTNR=7
XDG_SESSION_ID=c1
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
ANDROID_HOME=/home/seed/android/android-sdk-linux
GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1
TERM=xterm-256color
VTE_VERSION=4205
SHELL=/bin/bash
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libboost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0
WINDOWID=65011722
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1157
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=seed
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:01:cd=40;33:01:or=40;31:01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lzh=01;31:*.lzh=01;31:*.lzm=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.t7z=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
QT_ACCESSIBILITY=1
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/usr/share/upstart/xdg:/etc/xdg
DESKTOP_SESSION=ubuntu
PATH=/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Observation: The program ran and displayed the same information (environment variables) that it did previously.

Step 3: Next, I compared the difference of these two files using the `diff` command.

```
[02/26/19]seed@VM:~/.../Lab 3$ diff child parent
[02/26/19]seed@VM:~/.../Lab 3$
```

Conclusion: The child inherits the exact same environment that the parent has. This is evident when we run the `diff` command since it shows no differences between the files. These files are identical.

Task 3: Environment Variables and execve()

Step 1: I created a program called task3.c that will use the function execve(). This program simply executes a program called /usr/bin/env, which prints out the environment variables of the current process.

Program: task3.c

```
#include <stdio.h>
#include <stdlib.h>

extern char **environ;

int main()
{
    char *argv[2];

    argv[0] = "/usr/bin/env";
    argv[1] = NULL;

    execve("/usr/bin/env", argv, NULL);

    return 0;
}
```

```
[02/26/19]seed@VM:~/.../Lab 3$ ./task3
[02/26/19]seed@VM:~/.../Lab 3$
```

Observation: The output of the program is nothing since the 3rd argument in execve is NULL.

Step 2: Next, I changed the 3rd argument of execve() from NULL to environ.

```
#include <stdio.h>
#include <stdlib.h>

extern char **environ;

int main()
{
    char *argv[2];

    argv[0] = "/usr/bin/env";
    argv[1] = NULL;

    execve("/usr/bin/env", argv, environ);

    return 0;
}
```

Observation: The output displays the environment variables when the third argument is changed to environ.

Conclusion: The new program gets its environment variables by changing the 3rd argument from NULL to environ. When it's NULL, the environment variables are not inherited.

Task 4: Environment Variables and system()

In this task, we study how environment variables are affected when a new program is executed via the `system()` function. Unlike `execve()`, `system()` actually executes `"/bin/sh -c command"`, i.e., it executes `/bin/sh`, and asks the shell to execute the command.

Program: task4.c

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    system("/usr/bin/env");

    return 0;
}
```

```
[02/26/10]seed@VM:~/.../Lab 3$ gcc -o task4 task4.c
[02/26/10]seed@VM:~/.../Lab 3$ ./task4
LESSOPEN=| /usr/bin/lesspipe %s
GNOME KEYRING_PID=
USER=seed
LANGUAGE=en_US
UPSTART_INSTANCE=
J2SDKDIR=/usr/lib/jvm/java-8-oracle
XDG_SEAT=seat0
SESSION=ubuntu
XDG_SESSION_TYPE=x11
COMPIZ_CONFIG_PROFILE=ubuntu
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
SHLVL=1
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
HOME=/home/seed
QT4_IM_MODULE=xim
OLDPWD=/home/seed/Documents
DESKTOP_SESSION=ubuntu
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
GTK_MODULES=gail:atk-bridge:unity-gtk-module
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
INSTANCE=
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-H0q7s4p2b6
GNOME KEYRING CONTROL=
QT_QPA_PLATFORMTHEME=appmenu-qt5
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
IM_CONFIG_PHASE=1
SESSIONTYPE=gnome-session
UPSTART_JOB=unity7
LOGNAME=seed
GTK_IM_MODULE=ibus
WINDOWID=65011722
=./task4
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
XDG_SESSION_ID=c1
TERM=xterm-256color
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GTK2_MODULES=overlay-scrollbar
PATH=/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin:/home/seed/android/android-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-tools:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
GDM_LANG=en_US
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_RUNTIME_DIR=/run/user/1000
COMPIZ_BIN_PATH=/usr/bin/
DISPLAY=:0
LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libboost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0
LANG=en_US.UTF-8
XDG_CURRENT_DESKTOP=Unity
```

Observation: The `system()` function executed `/usr/bin/env`.

Task 5: Environment Variable and Set-UID Programs

Step 1: I wrote the provided program and saved it as task5.c.

Program: task5.c

```
#include <stdio.h>
#include <stdlib.h>

extern char **environ;

void main()
{
    int i = 0;
    while (environ[i] != NULL) {
        printf("%s\n", environ[i]);
        i++;
    }
}
```

Step 2: I compiled this program, changed the ownership to root, and made it a Set-UID program.
(Screenshot of compilation command was lost).

```
[02/26/19]seed@VM:~/.../Lab 3$ sudo chown root task5
[02/26/19]seed@VM:~/.../Lab 3$ sudo chmod 4755 task5
[02/26/19]seed@VM:~/.../Lab 3$ ./task5
```

Step 3: In my user shell, I used the export command to set the following environment variables PATH, LD_LIBRARY_PATH, .

```
[02/26/19]seed@VM:~/.../Lab 3$ export PATH=/home/seed:$PATH
[02/26/19]seed@VM:~/.../Lab 3$ printenv PATH
/home/seed:/home/seed:/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin:/home/seed/android/android-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-tools:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
[02/26/19]seed@VM:~/.../Lab 3$ printenv LD_LIBRARY_PATH
/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
[02/26/19]seed@VM:~/.../Lab 3$ export LD_LIBRARY_PATH=/home/seed:$LD_LIBRARY_PATH
[02/26/19]seed@VM:~/.../Lab 3$ printenv LD_LIBRARY_PATH/home/seed:/home/seed/source
[02/26/19]seed@VM:~/.../Lab 3$ printenv LD_LIBRARY_PATH
/home/seed:/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
[02/26/19]seed@VM:~/.../Lab 3$ printenv task5
[02/26/19]seed@VM:~/.../Lab 3$ export task5='task5 env variable'
[02/26/19]seed@VM:~/.../Lab 3$ printenv task5
task5 env variable
[02/26/19]seed@VM:~/.../Lab 3$ ./setuidenv > setuidenv_result
bash: ./setuidenv: No such file or directory
[02/26/19]seed@VM:~/.../Lab 3$ ./task5 > task5_result
[02/26/19]seed@VM:~/.../Lab 3$
[02/26/19]seed@VM:~/.../Lab 3$ ./task5 > task5_result
[02/26/19]seed@VM:~/.../Lab 3$
[02/26/19]seed@VM:~/.../Lab 3$ env > task5_result
[02/26/19]seed@VM:~/.../Lab 3$
[02/26/19]seed@VM:~/.../Lab 3$ env > env_result
[02/26/19]seed@VM:~/.../Lab 3$
[02/26/19]seed@VM:~/.../Lab 3$ diff task5_result env_result
[02/26/19]seed@VM:~/.../Lab 3$ printenv task5
task5 env variable
```

Observation: After running the program, the LD_LIBRARY_PATH was not displayed.

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

First, I was asked to type in

```
$ export PATH=/home/seed:$PATH
```

The program below (task6.c) is supposed to execute the `/bin/ls` command; however, the programmer only uses the relative path for the `ls` command, rather than the absolute path.

```
int main()
{
    system("ls");
    return 0;
}
```

```
[02/27/19]seed@VM:~/.../Lab 3$ gcc task6.c -o task6
task6.c: In function 'main':
task6.c:3:2: warning: implicit declaration of function 'system' [-Wimplicit-function-declaratio
n]
   system("ls");
   ^
[02/27/19]seed@VM:~/.../Lab 3$ sudo chown root task6
[02/27/19]seed@VM:~/.../Lab 3$ sudo chmod 4755 task6
[02/27/19]seed@VM:~/.../Lab 3$ ls -l task6
-rwsr-xr-x 1 root seed 7348 Feb 27 17:55 task6
[02/27/19]seed@VM:~/.../Lab 3$ gcc ls.c -o ls
gcc: error: ls.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
[02/27/19]seed@VM:~/.../Lab 3$ task6
child      parent      task2      task3      task4      task5      task6
env_result setuidenv_result task2.c    task3.c    task4.c    task5.c    task6.c
[02/27/19]seed@VM:~/.../Lab 3$ /bin/ls
child      parent      task2      task3      task4      task5      task6
env_result setuidenv_result task2.c    task3.c    task4.c    task5.c    task6.c
[02/27/19]seed@VM:~/.../Lab 3$
```

Observation: I compiled the program, changed the owner to root, and made it a Set-UID program. The screenshot above shows that executing `task6` displayed the same contents that `/bin/ls` displays and therefore, Set-UID can run malicious programs with root privileges if `PATH` is changed.

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

Step 1: First, we will see how these environment variables influence the behavior of dynamic loader/linker when running a normal program. Please follow these steps:

1. Let us build a dynamic link library. Create the following program, and name it mylib.c. It basically overrides the sleep() function in libc:

```
#include <stdio.h>
void sleep (int s)
{
    /* If this is invoked by a privileged program,
       you can do damages here! */
    printf("I am not sleeping!\n");
}
```

2. We can compile the above program using the following commands:

```
% gcc -fPIC -g -c mylib.c
```

```
% gcc -shared -o libmylib.so.1.0.1 mylib.o -lc
```

```
[02/27/19]seed@VM:~/.../Lab 3$ gcc -o myprog myprog.c
myprog.c: In function 'main':
myprog.c:4:2: warning: implicit declaration of function 'sleep' [-Wimplicit-function-declaratio
n]
    sleep(1);
    ^
[02/27/19]seed@VM:~/.../Lab 3$ myprog
[02/27/19]seed@VM:~/.../Lab 3$ gcc -fPIC -g -c mylib.c
[02/27/19]seed@VM:~/.../Lab 3$ gcc -shared -o libmylib.so.1.0.1 mylib.o -lc
```

3. Now, set the LD_PRELOAD environment variable:

```
% export LD_PRELOAD=./libmylib.so.1.0.1
```

```
[02/27/19]seed@VM:~/.../Lab 3$ export LD_PRELOAD=./libmylib.so.1.0.1
```

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

```
[02/27/19]seed@VM:~/.../Lab 3$ gcc -o myprog myprog.c
```

Step 2: After you have done the above, please run myprog under the following conditions, and observe what happens.

- Make myprog a regular program, and run it as a normal user.

```
[02/27/19]seed@VM:~/.../Lab 3$ gcc -o myprog myprog.c
myprog.c: In function 'main':
myprog.c:4:2: warning: implicit declaration of function 'sleep' [-Wimplicit-function-declaration]
    sleep(1);
    ^
[02/27/19]seed@VM:~/.../Lab 3$ myprog
I am not sleeping!
```

- Make myprog a Set-UID root program, and run it as a normal user.

```
[02/27/19]seed@VM:~/.../Lab 3$ sudo chown root myprog
[sudo] password for seed:
[02/27/19]seed@VM:~/.../Lab 3$ sudo chmod 4755 myprog
[02/27/19]seed@VM:~/.../Lab 3$ myprog
```

- Make myprog a Set-UID root program, export the LD PRELOAD environment variable again in the root account and run it.

```
[02/27/19]seed@VM:~/.../Lab 3$ su
Password:
root@VM:/home/seed/Documents/Lab 3# export LD_PRELOAD=./libmylib.so.1.0.1
root@VM:/home/seed/Documents/Lab 3# myprog
I am not sleeping!
```

- Make myprog a Set-UID user1 program (i.e., the owner is user1, which is another user account), export the LD PRELOAD environment variable again in a different user's account (not-root user) and run it.

```
root@VM:/home/seed/Documents/Lab 3# sudo adduser Steve
adduser: Please enter a username matching the regular expression configured
via the NAME_REGEX[SYSTEM] configuration variable. Use the '--force-badname'
option to relax this check or reconfigure NAME_REGEX.
root@VM:/home/seed/Documents/Lab 3# sudo adduser steve
Adding user `steve' ...
Adding new group `steve' (1001) ...
Adding new user `steve' (1001) with group `steve' ...
Creating home directory `/home/steve' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for steve
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
root@VM:/home/seed/Documents/Lab 3# sudo chown steve myprog
root@VM:/home/seed/Documents/Lab 3# export LD_PRELOAD=./libmylib.so.1.0.1
root@VM:/home/seed/Documents/Lab 3# myprog
I am not sleeping!
```

Observations: In the first scenario, the program runs as normal because the permissions have not yet been changed. In the second scenario, the program is made to be a root-owned Set-UID program and is run by a normal user, but there is no output from the program. In the third scenario, I exported the LD_PRELOAD variable and ran the program successfully. Though on second thought, I'm not sure if I was supposed to stick with root while executing. In the last scenario, I assigned the program to the user 'steve' and exported the LD_PRELOAD variable again and ran the program successfully. Again, I'm not sure that keeping root access was the right thing to do. It seems that normal users are unable to run the program but root users can. But, I speculate that, according to the task requirements, that the LD_PRELOAD variable being exported will allow the root-owned Set-UID program to run by a normal user or a different user. I don't believe I completed this task perfectly, but I think that's what it was trying to get us to do.

Task 8: Invoking External Programs using system() versus execve()

Step 1: I compiled the program task8.c and made it a root-owned Set-UID program. The program will use system() to invoke the command. If you were Bob, can you compromise the integrity of the system? For example, can you remove a file that is not writable to you?

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {

    char *v[3];
    char *command;

    if(argc < 2) {
        printf("Please type a file name.\n");
        return 1;
    }

    v[0] = "/bin/cat"; v[1] = argv[1]; v[2] = NULL;
    command = malloc(strlen(v[0]) + strlen(v[1]) + 2);
    sprintf(command, "%s %s", v[0], v[1]);

    // Use only one of the following
    system(command);
    //execve(v[0], v, NULL);

    return 0;
}
```

```

[02/27/19]seed@VM:~/.../Lab 3$ subl
[02/27/19]seed@VM:~/.../Lab 3$ gcc -o task8 task8.c
[02/27/19]seed@VM:~/.../Lab 3$ sudo chown root task8
[02/27/19]seed@VM:~/.../Lab 3$ sudo chmod 4755 task8
[02/27/19]seed@VM:~/.../Lab 3$ subl deleteme.txt
[02/27/19]seed@VM:~/.../Lab 3$ sudo chown root:root deleteme.txt
[02/27/19]seed@VM:~/.../Lab 3$ /bin/ls -l deleteme.txt
-rw-rw-r-- 1 root root 18 Feb 27 21:34 deleteme.txt
[02/27/19]seed@VM:~/.../Lab 3$ su steve
Password:
steve@VM:/home/seed/Documents/Lab 3$ ./task8 "deleteme.txt;rm deleteme.txt"
Can you delete me?steve@VM:/home/seed/Documents/Lab 3$ cat deleteme.txt
cat: deleteme.txt: No such file or directory
steve@VM:/home/seed/Documents/Lab 3$ ls -l deleteme.txt
ls: cannot access 'deleteme.txt': No such file or directory
steve@VM:/home/seed/Documents/Lab 3$ ls
child          mylib.c  myprog.c      task2         task3.c  task5         task6.c
env_result     mylib.o  parent        task2.c      task4    task5.c      task8
libmylib.so.1.0.1 myprog  setuidenv_result task3         task4.c  task6         task8.c
steve@VM:/home/seed/Documents/Lab 3$ exit
exit
[02/27/19]seed@VM:~/.../Lab 3$ ls
child          mylib.c  myprog.c      task2         task3.c  task5         task6.c
env_result     mylib.o  parent        task2.c      task4    task5.c      task8
libmylib.so.1.0.1 myprog  setuidenv_result task3         task4.c  task6         task8.c
[02/27/19]seed@VM:~/.../Lab 3$ /bin/ls -l
total 116
-rw-rw-r-- 1 seed seed 4019 Feb 26 10:24 child
-rw-rw-r-- 1 seed seed 4080 Feb 26 12:02 env_result
-rwxrwxr-x 1 seed seed 7928 Feb 27 18:24 libmylib.so.1.0.1
-rw-rw-r-- 1 seed seed 157 Feb 27 18:24 mylib.c
-rw-rw-r-- 1 seed seed 2588 Feb 27 18:24 mylib.o
-rwsr-xr-x 1 steve seed 7348 Feb 27 18:24 myprog
-rw-rw-r-- 1 seed seed 53 Feb 27 18:23 myprog.c
-rw-rw-r-- 1 seed seed 4019 Feb 26 10:26 parent
-rw-rw-r-- 1 seed seed 3801 Feb 26 12:02 setuidenv_result
-rwxrwxr-x 1 seed seed 7496 Feb 26 10:26 task2
-rwxrw-r-- 1 seed seed 368 Feb 26 10:26 task2.c
-rwxrwxr-x 1 seed seed 7448 Feb 26 11:31 task3
-rw-rw-r-- 1 seed seed 194 Feb 26 11:30 task3.c
-rwxrwxr-x 1 seed seed 7348 Feb 26 11:39 task4
-rw-rw-r-- 1 seed seed 92 Feb 26 11:39 task4.c
-rwsr-xr-x 1 root seed 7396 Feb 26 11:42 task5
-rw-rw-r-- 1 seed seed 162 Feb 26 11:42 task5.c
-rwsr-xr-x 1 root seed 7348 Feb 27 17:55 task6
-rw-rw-r-- 1 seed seed 41 Feb 27 17:44 task6.c
-rwsr-xr-x 1 root seed 7544 Feb 27 21:52 task8
-rw-rw-r-- 1 seed seed 432 Feb 27 21:52 task8.c
[02/27/19]seed@VM:~/.../Lab 3$

```

Observation: Using the system() command I was able to delete the root-owned Set-UID text file that I created and named 'deleteme.txt' using the program.

Step 2: Comment out the `system(command)` statement, and uncomment the `execve()` statement in the program from Step 1; 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.

```
[02/27/19]seed@VM:~/.../Lab 3$ gcc -o task8 task8.c
task8.c: In function 'main':
task8.c:21:2: warning: implicit declaration of function 'execve' [-Wimplicit-function-declaration]
   execve(v[0], v, NULL);
   ^
[02/27/19]seed@VM:~/.../Lab 3$ sudo chown root task8
[02/27/19]seed@VM:~/.../Lab 3$ sudo chmod 4755 task8
[02/27/19]seed@VM:~/.../Lab 3$ sudo chown root:root deleteme.txt
[02/27/19]seed@VM:~/.../Lab 3$ su steve
Password:
steve@VM:/home/seed/Documents/Lab 3$ ./task8 deleteme.txt;rm deleteme.txt
Can you delete me?rm: remove write-protected regular file 'deleteme.txt'? y
rm: cannot remove 'deleteme.txt': Permission denied
steve@VM:/home/seed/Documents/Lab 3$
```

Observations: The changes were made to the program from Step 1 and the user 'steve' attempted to use the program to delete the `deleteme.txt` file and was unable to do so. This shows that the `system()` command will execute the `'rm deleteme.txt'` portion but the `execve()` command will not.

Task 9: Capability Leaking

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

void main()
{ int fd;

    /* Assume that /etc/zxx is an important sy
     * and it is owned by root with permission
     * Before running this program, you should
     * the file /etc/zxx first. */
    fd = open("/etc/zxx", O_RDWR | O_APPEND);
    if (fd == -1) {
        printf("Cannot open /etc/zxx\n");
        exit(0);
    }

    /* Simulate the tasks conducted by the pro
    sleep (1);

    /* After the task, the root privileges are
     it's time to relinquish the root privil
    setuid(getuid()); /* getuid() returns the

    if (fork()) { /* In the parent process */
        close (fd);
        exit(0);
    } else { /* in the child process*/
        /* Now, assume that the child process
         attackers have injected the follow

        write (fd, "Malicious Data\n", 15);
        close (fd);
    }
}
```



```

[02/27/19]seed@VM:~/.../Lab 3$ subl task9.c
[02/27/19]seed@VM:~/.../Lab 3$ gcc -o task9 task9.c
task9.c: In function 'main':
task9.c:19:2: warning: implicit declaration of function 'sleep' [-Wimplicit-function-declaration]
    sleep(1);
    ^
task9.c:23:2: warning: implicit declaration of function 'setuid' [-Wimplicit-function-declaration]
    setuid(getuid()); /* getuid() returns the real uid */
    ^
task9.c:23:9: warning: implicit declaration of function 'getuid' [-Wimplicit-function-declaration]
    setuid(getuid()); /* getuid() returns the real uid */
    ^
task9.c:25:6: warning: implicit declaration of function 'fork' [-Wimplicit-function-declaration]
    if (fork()) { /* In the parent process */
    ^
task9.c:26:3: warning: implicit declaration of function 'close' [-Wimplicit-function-declaration]
    close(fd);
    ^
task9.c:32:3: warning: implicit declaration of function 'write' [-Wimplicit-function-declaration]
    write(fd, "Malicious Data\n", 15);
    ^

[02/27/19]seed@VM:~/.../Lab 3$ sudo chown root task9
[02/27/19]seed@VM:~/.../Lab 3$ sudo chmod 4755 task9
[02/27/19]seed@VM:~/.../Lab 3$ /bin/ls -l task9
-rwsr-xr-x 1 root seed 7640 Feb 27 22:13 task9
[02/27/19]seed@VM:~/.../Lab 3$ sudo su root
root@VM:/home/seed/Documents/Lab 3# cd /etc
root@VM:/etc# exit
exit
[02/27/19]seed@VM:~/.../Lab 3$ ./task9
Cannot open /etc/zzz
[02/27/19]seed@VM:~/.../Lab 3$ cd /etc
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.
[02/27/19]seed@VM:/etc$ ls
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.
acpi                gai.conf            lsb-release         sane.d
adduser.conf         gconf               ltrace.conf         securetty
alternatives         gdb                 machine-id           security
anacrontab           ghostscript         magic                selinux
apache2              gnome               magic.mime           sensors3.conf
apg.conf             gnome-app-install   mailcap              sensors.d
apm                  gnome-vfs-2.0       mailcap.order        services
apparmor             groff               manpath.config       sgml
apparmor.d           group               mime.types           shadow
appport             group-              mke2fs.conf          shadow-

```

```

root@VM: /etc
root@VM:/etc# subl zzz
root@VM:/etc# cat zzz
Paragraphs
Paragraphs
Important stuff
Don't read this stuff please
root@VM:/etc# /bin/ls -l zzz
-rw-r--r-- 1 root root 66 Feb 27 22:21 zzz
root@VM:/etc# exit
exit
root@VM:/etc# exit
exit
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open sha
red object file): ignored.
[02/27/19]seed@VM:/etc$ cd /home/seed/"Lab 3"
bash: cd: /home/seed/Lab 3: No such file or directory
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open sha
red object file): ignored.
[02/27/19]seed@VM:/etc$ cd /home
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open sha
red object file): ignored.
[02/27/19]seed@VM:/home$ cd home
bash: cd: home: No such file or directory
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open sha
red object file): ignored.
[02/27/19]seed@VM:/home$ cd seed
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open sha
red object file): ignored.
[02/27/19]seed@VM:~$ cd /seed
bash: cd: /seed: No such file or directory
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open sha
red object file): ignored.
[02/27/19]seed@VM:~$ ls
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open sha
red object file): ignored.
android Customization Documents examples.desktop Music Public Templates
bin Desktop Downloads lib Pictures source Videos
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open sha
red object file): ignored.
[02/27/19]seed@VM:~$ cd Documents
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open sha
red object file): ignored.
[02/27/19]seed@VM:~/Documents$ ls
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open sha
red object file): ignored.
Lab 3
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open sha
red object file): ignored.
[02/27/19]seed@VM:~/Documents$ cd "Lab 3"
[02/27/19]seed@VM:~/.../Lab 3$ ls
child          mylib.c  myprog.c      task2      task3.c  task5      task6.c  task9
env_result     mylib.o  parent        task2.c    task4     task5.c    task8     task9.c
libmylib.so.1.0.1 myprog  setuidenv result task3      task4.c  task6     task8.c  Task9.c
[02/27/19]seed@VM:~/.../Lab 3$ ./task9
[02/27/19]seed@VM:~/.../Lab 3$ cat zzz
cat: zzz: No such file or directory
[02/27/19]seed@VM:~/.../Lab 3$ cat /etc/zzz
Paragraphs
Paragraphs
Important stuff
Don't read this stuff please
Malicious Data
[02/27/19]seed@VM:~/.../Lab 3$

```

Observations: After writing the program, I created the file “zzz.txt” and made it a root-owned Set-UID program. This task shows that when revoking privileges, sometimes the privileged capabilities are left behind with the program. This is called capability leaking. I believe that the capabilities are passed from the parent to the child in the fork process, but it’s a little above my head so that’s my best guess. The zzz.txt file will be modified with the additional text “Malicious Data” at the end, as seen in the last screenshot.