EE-3233-01T-Summer 2025-Systems Programming for Engineers

Assignment 2: Exam1

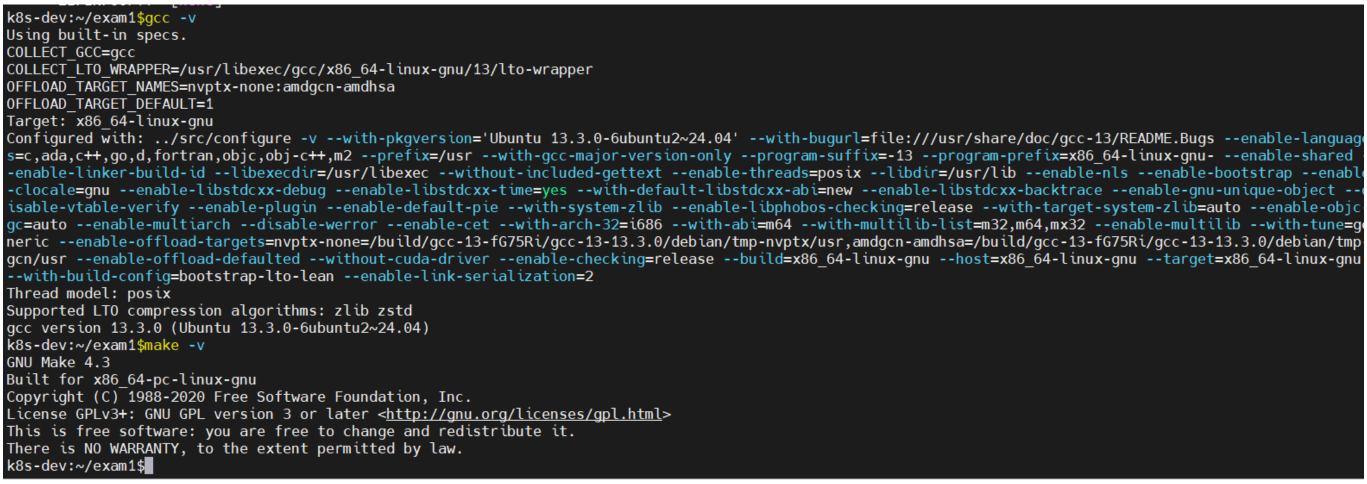
Name: **Arnav Gupta**

Utsa abcID : **Enp615**

Developer Machine Setup : vscode for c/c++ on ubuntu

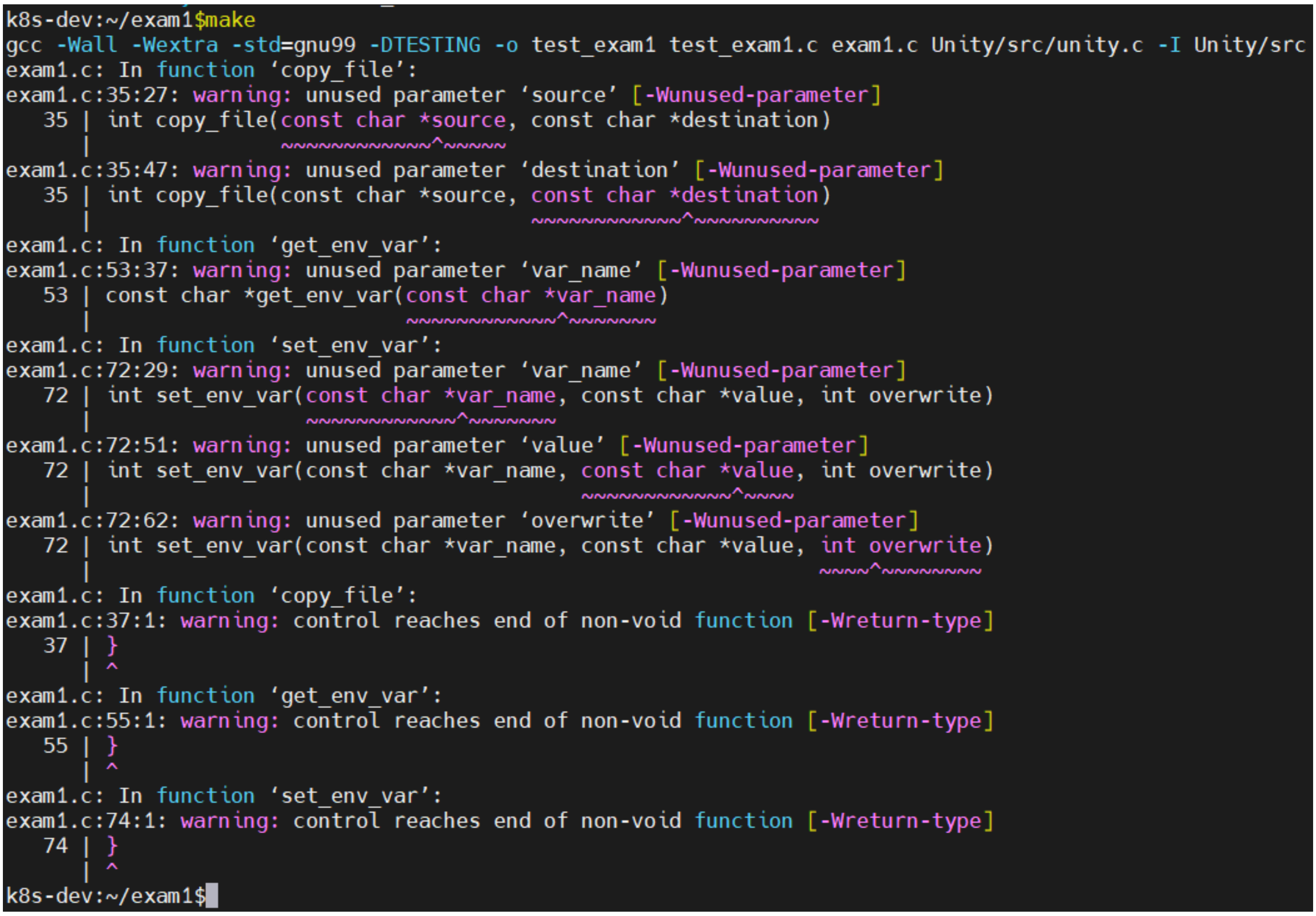
==============================

- Install VSCode
- sudo apt update -y && sudo apt upgrade -y
- sudo apt install build-essential

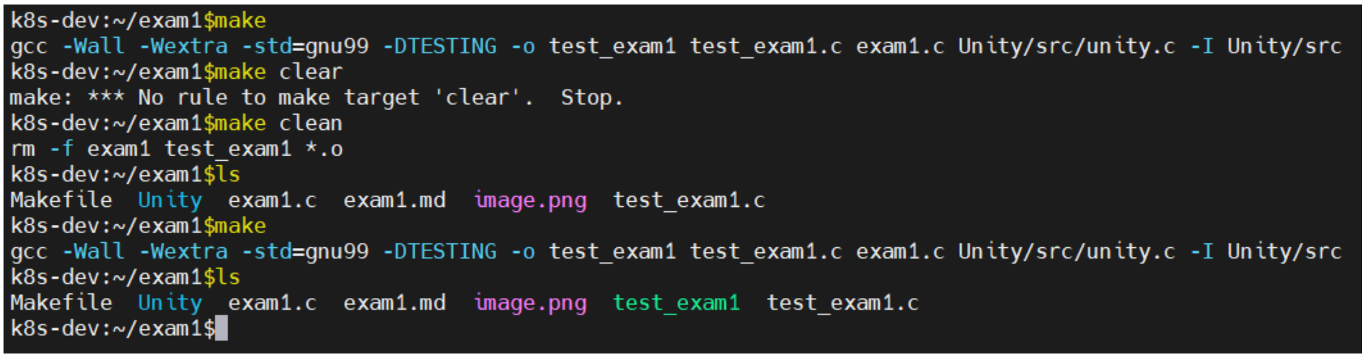


```
k8s-dev:~/exam1$gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-linux-gnu/13/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none:amdgcn-amdhsa
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 13.3.0-6ubuntu2~24.04' --with-bugurl=file:///usr/share/doc/gcc-13/README.Bugs --enable-languag
s=c,ada,c++,go,d,fortran,objc,obj-c++,m2 --prefix=/usr --with-gcc-major-version-only --program-suffix=-13 --program-prefix=x86_64-linux-gnu- --enable-shared
-enable-linker-build-id --libexecdir=/usr/libexec --without-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --enable-bootstrap --enabl
-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-libstdcxx-backtrace --enable-gnu-unique-object --
isable-vtable-verify --enable-plugin --enable-default-pie --with-system-zlib --enable-libphobos-checking=release --with-target-system-zlib=auto --enable-objc
gc=auto --enable-multiarch --disable-werror --enable-cet --with-arch-32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=g
neric --enable-offload-targets=nvptx-none=/build/gcc-13-fG75Ri/gcc-13-13.3.0/debian/tmp-nvptx/usr,amdgcn-amdhsa=/build/gcc-13-fG75Ri/gcc-13-13.3.0/debian/tmp
gcn/usr --enable-offload-defaulted --without-cuda-driver --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu
--with-build-config=bootstrap-lto-lean --enable-link-serialization=2
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 13.3.0 (Ubuntu 13.3.0-6ubuntu2~24.04)
k8s-dev:~/exam1$make -v
GNU Make 4.3
Built for x86_64-pc-linux-gnu
Copyright (C) 1988-2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
k8s-dev:~/exam1$
```

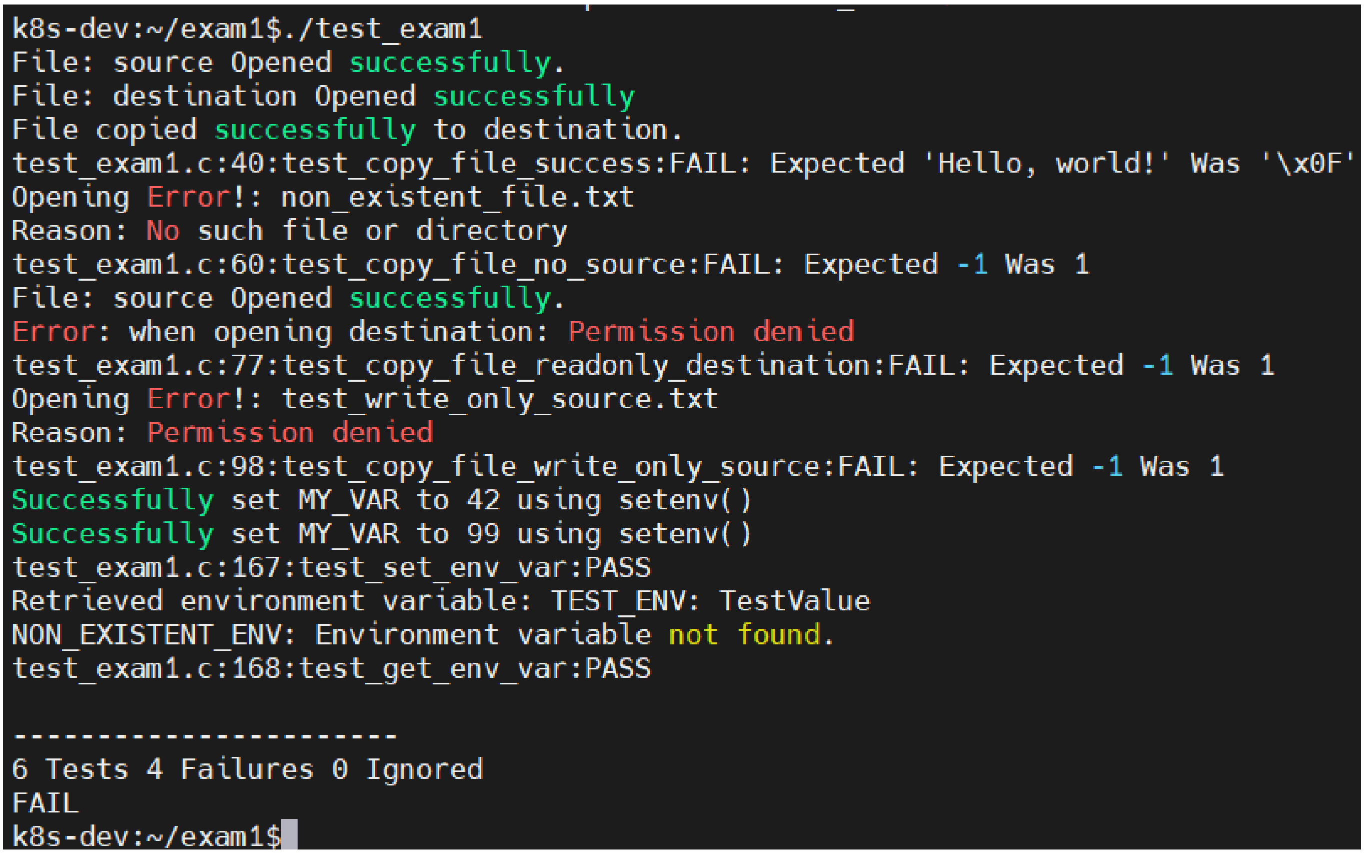
##Execution steps



```
k8s-dev:~/exam1$make
gcc -Wall -Wextra -std=gnu99 -DTESTING -o test_exam1 test_exam1.c exam1.c Unity/src/unity.c -I Unity/src
exam1.c: In function ‘copy_file’:
exam1.c:35:27: warning: unused parameter ‘source’ [-Wunused-parameter]
   35 | int copy_file(const char *source, const char *destination)
      |               ~~~~~~~~~~~~^~~~~~
exam1.c:35:47: warning: unused parameter ‘destination’ [-Wunused-parameter]
   35 | int copy_file(const char *source, const char *destination)
      |                                   ~~~~~~~~~~~~^~~~~~~~~~~
exam1.c: In function ‘get_env_var’:
exam1.c:53:37: warning: unused parameter ‘var_name’ [-Wunused-parameter]
   53 | const char *get_env_var(const char *var_name)
      |                         ~~~~~~~~~~~~^~~~~~~~
exam1.c: In function ‘set_env_var’:
exam1.c:72:29: warning: unused parameter ‘var_name’ [-Wunused-parameter]
   72 | int set_env_var(const char *var_name, const char *value, int overwrite)
      |                 ~~~~~~~~~~~~^~~~~~~~
exam1.c:72:51: warning: unused parameter ‘value’ [-Wunused-parameter]
   72 | int set_env_var(const char *var_name, const char *value, int overwrite)
      |                                       ~~~~~~~~~~~~^~~~~
exam1.c:72:62: warning: unused parameter ‘overwrite’ [-Wunused-parameter]
   72 | int set_env_var(const char *var_name, const char *value, int overwrite)
      |                                                          ~~~~^~~~~~~~~
exam1.c: In function ‘copy_file’:
exam1.c:37:1: warning: control reaches end of non-void function [-Wreturn-type]
   37 | }
      | ^
exam1.c: In function ‘get_env_var’:
exam1.c:55:1: warning: control reaches end of non-void function [-Wreturn-type]
   55 | }
      | ^
exam1.c: In function ‘set_env_var’:
exam1.c:74:1: warning: control reaches end of non-void function [-Wreturn-type]
   74 | }
      | ^
k8s-dev:~/exam1$
```

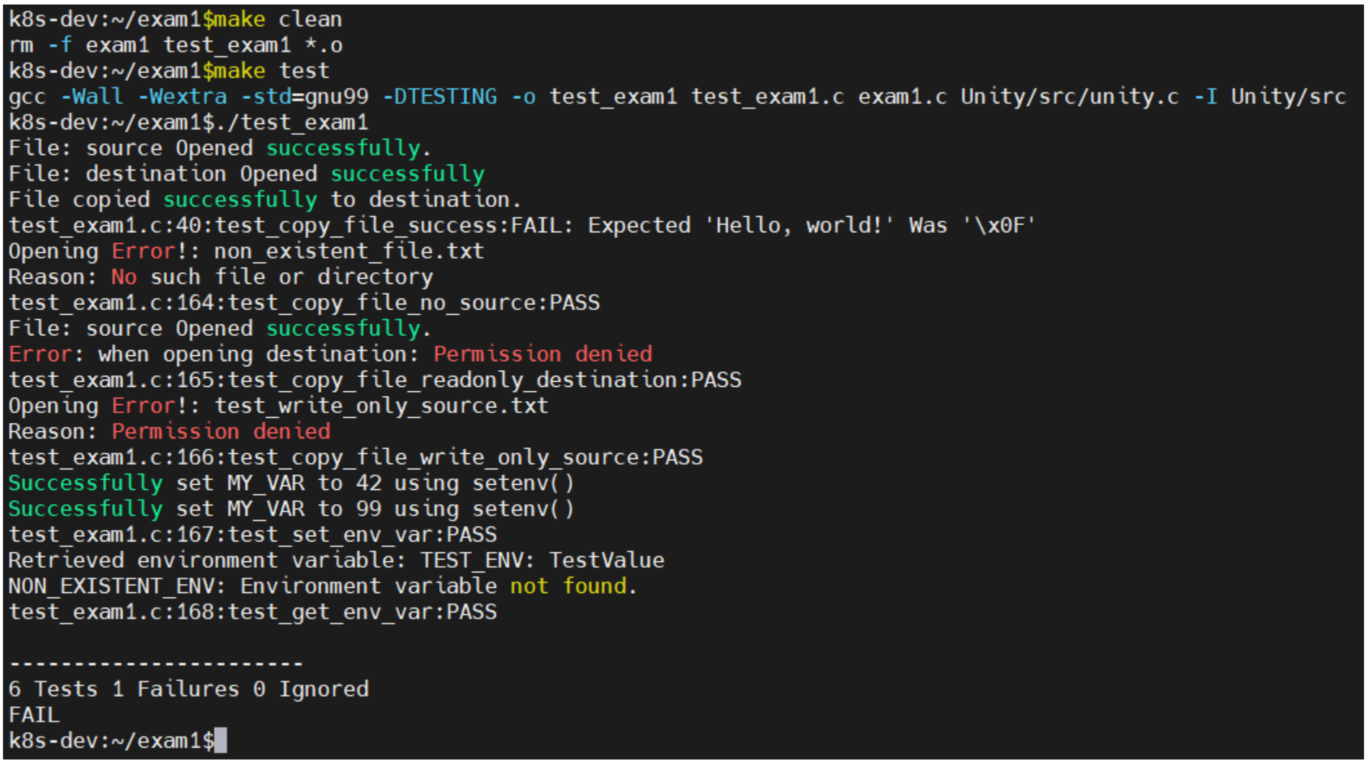


```
k8s-dev:~/exam1$make
gcc -Wall -Wextra -std=gnu99 -DTESTING -o test_exam1 test_exam1.c exam1.c Unity/src/unity.c -I Unity/src
k8s-dev:~/exam1$make clear
make: *** No rule to make target 'clear'.  Stop.
k8s-dev:~/exam1$make clean
rm -f exam1 test_exam1 *.o
k8s-dev:~/exam1$ls
Makefile  Unity  exam1.c  exam1.md  image.png  test_exam1.c
k8s-dev:~/exam1$make
gcc -Wall -Wextra -std=gnu99 -DTESTING -o test_exam1 test_exam1.c exam1.c Unity/src/unity.c -I Unity/src
k8s-dev:~/exam1$ls
Makefile  Unity  exam1.c  exam1.md  image.png  test_exam1  test_exam1.c
k8s-dev:~/exam1$
```



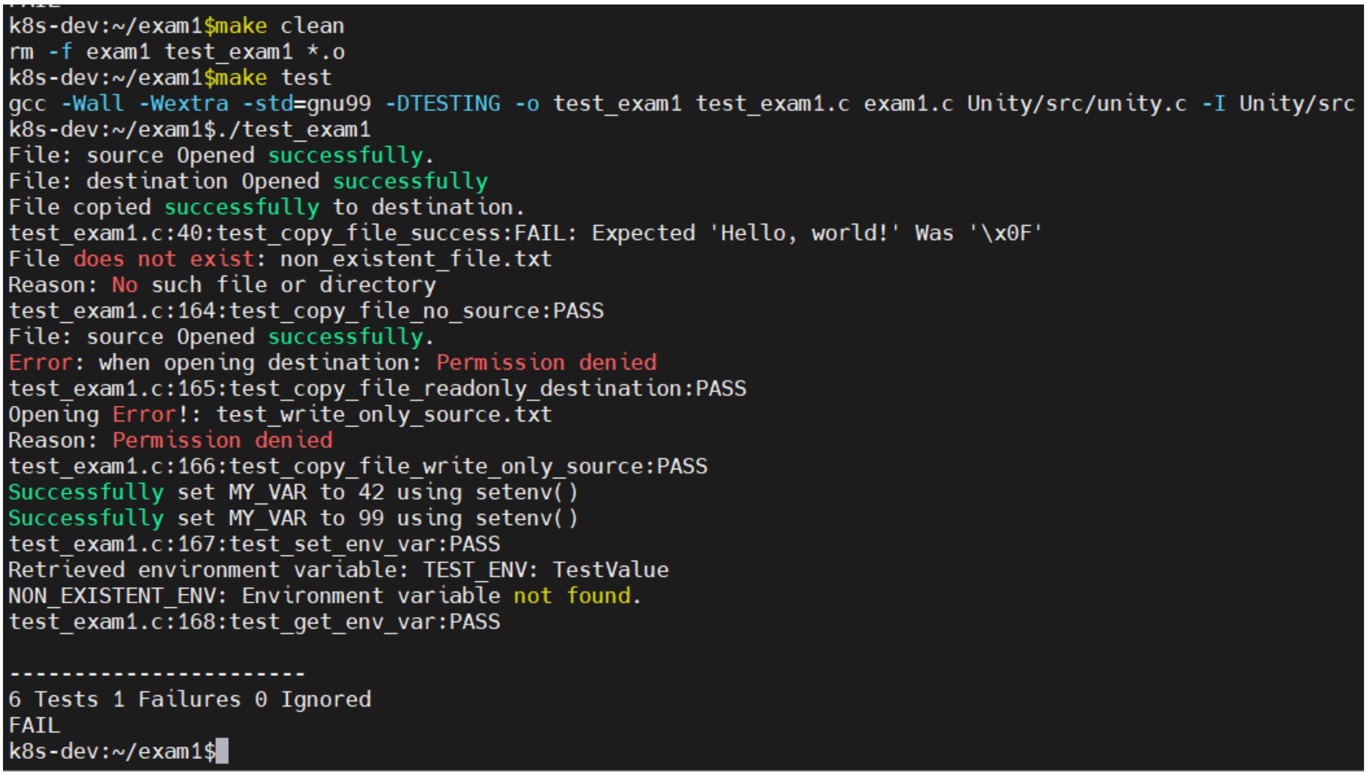
```
k8s-dev:~/exam1$./test_exam1
File: source Opened successfully.
File: destination Opened successfully
File copied successfully to destination.
test_exam1.c:40:test_copy_file_success:FAIL: Expected 'Hello, world!' Was '\x0F'
Opening Error!: non_existent_file.txt
Reason: No such file or directory
test_exam1.c:60:test_copy_file_no_source:FAIL: Expected -1 Was 1
File: source Opened successfully.
Error: when opening destination: Permission denied
test_exam1.c:77:test_copy_file_readonly_destination:FAIL: Expected -1 Was 1
Opening Error!: test_write_only_source.txt
Reason: Permission denied
test_exam1.c:98:test_copy_file_write_only_source:FAIL: Expected -1 Was 1
Successfully set MY_VAR to 42 using setenv()
Successfully set MY_VAR to 99 using setenv()
test_exam1.c:167:test_set_env_var:PASS
Retrieved environment variable: TEST_ENV: TestValue
NON_EXISTENT_ENV: Environment variable not found.
test_exam1.c:168:test_get_env_var:PASS

-----------------------
6 Tests 4 Failures 0 Ignored
FAIL
k8s-dev:~/exam1$
```



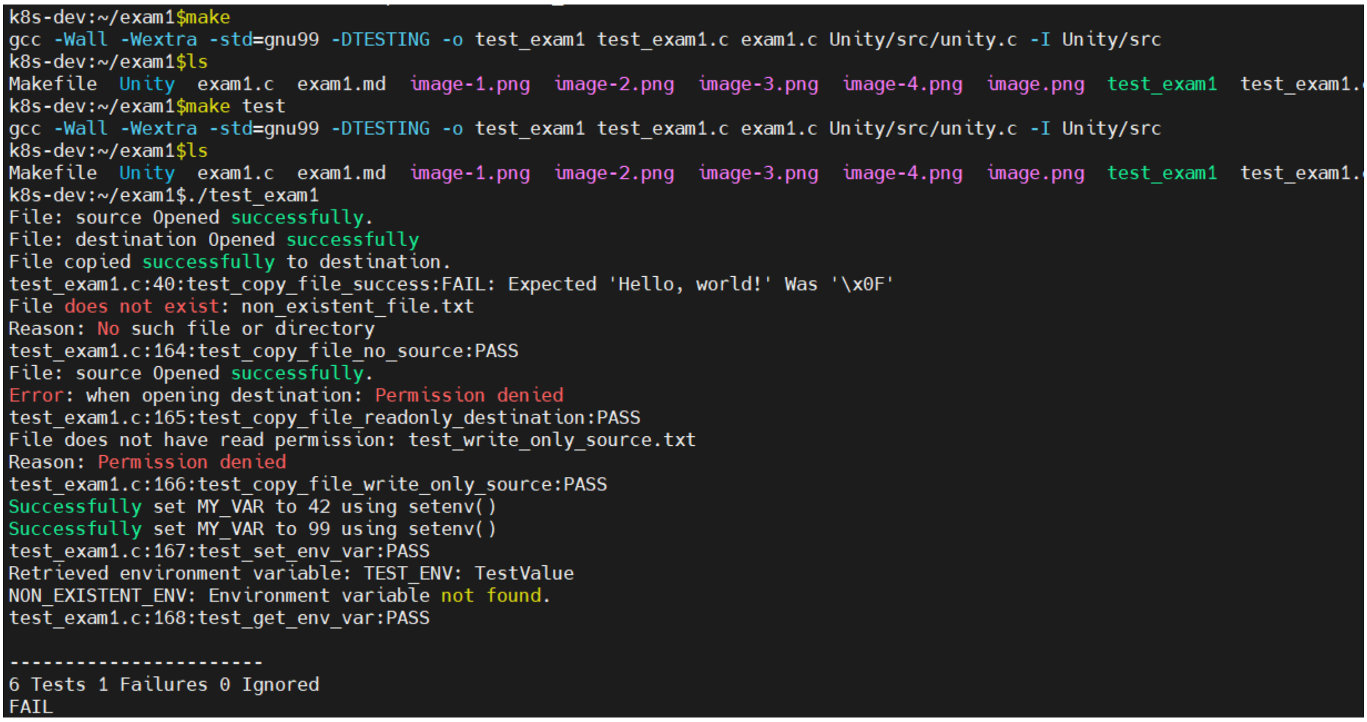
```
k8s-dev:~/exam1$make clean
rm -f exam1 test_exam1 *.o
k8s-dev:~/exam1$make test
gcc -Wall -Wextra -std=gnu99 -DTESTING -o test_exam1 test_exam1.c exam1.c Unity/src/unity.c -I Unity/src
k8s-dev:~/exam1$./test_exam1
File: source Opened successfully.
File: destination Opened successfully
File copied successfully to destination.
test_exam1.c:40:test_copy_file_success:FAIL: Expected 'Hello, world!' Was '\x0F'
Opening Error!: non_existent_file.txt
Reason: No such file or directory
test_exam1.c:164:test_copy_file_no_source:PASS
File: source Opened successfully.
Error: when opening destination: Permission denied
test_exam1.c:165:test_copy_file_readonly_destination:PASS
Opening Error!: test_write_only_source.txt
Reason: Permission denied
test_exam1.c:166:test_copy_file_write_only_source:PASS
Successfully set MY_VAR to 42 using setenv()
Successfully set MY_VAR to 99 using setenv()
test_exam1.c:167:test_set_env_var:PASS
Retrieved environment variable: TEST_ENV: TestValue
NON_EXISTENT_ENV: Environment variable not found.
test_exam1.c:168:test_get_env_var:PASS

-----------------------
6 Tests 1 Failures 0 Ignored
FAIL
k8s-dev:~/exam1$
```



```
k8s-dev:~/exam1$make clean
rm -f exam1 test_exam1 *.o
k8s-dev:~/exam1$make test
gcc -Wall -Wextra -std=gnu99 -DTESTING -o test_exam1 test_exam1.c exam1.c Unity/src/unity.c -I Unity/src
k8s-dev:~/exam1$./test_exam1
File: source Opened successfully.
File: destination Opened successfully
File copied successfully to destination.
test_exam1.c:40:test_copy_file_success:FAIL: Expected 'Hello, world!' Was '\x0F'
File does not exist: non_existent_file.txt
Reason: No such file or directory
test_exam1.c:164:test_copy_file_no_source:PASS
File: source Opened successfully.
Error: when opening destination: Permission denied
test_exam1.c:165:test_copy_file_readonly_destination:PASS
Opening Error!: test_write_only_source.txt
Reason: Permission denied
test_exam1.c:166:test_copy_file_write_only_source:PASS
Successfully set MY_VAR to 42 using setenv()
Successfully set MY_VAR to 99 using setenv()
test_exam1.c:167:test_set_env_var:PASS
Retrieved environment variable: TEST_ENV: TestValue
NON_EXISTENT_ENV: Environment variable not found.
test_exam1.c:168:test_get_env_var:PASS

-----------------------
6 Tests 1 Failures 0 Ignored
FAIL
k8s-dev:~/exam1$
```



```
k8s-dev:~/exam1$make
gcc -Wall -Wextra -std=gnu99 -DTESTING -o test_exam1 test_exam1.c exam1.c Unity/src/unity.c -I Unity/src
k8s-dev:~/exam1$ls
Makefile  Unity  exam1.c  exam1.md  image-1.png  image-2.png  image-3.png  image-4.png  image.png  test_exam1  test_exam1.
k8s-dev:~/exam1$make test
gcc -Wall -Wextra -std=gnu99 -DTESTING -o test_exam1 test_exam1.c exam1.c Unity/src/unity.c -I Unity/src
k8s-dev:~/exam1$ls
Makefile  Unity  exam1.c  exam1.md  image-1.png  image-2.png  image-3.png  image-4.png  image.png  test_exam1  test_exam1.
k8s-dev:~/exam1$./test_exam1
File: source Opened successfully.
File: destination Opened successfully
File copied successfully to destination.
test_exam1.c:40:test_copy_file_success:FAIL: Expected 'Hello, world!' Was '\x0F'
File does not exist: non_existent_file.txt
Reason: No such file or directory
test_exam1.c:164:test_copy_file_no_source:PASS
File: source Opened successfully.
Error: when opening destination: Permission denied
test_exam1.c:165:test_copy_file_readonly_destination:PASS
File does not have read permission: test_write_only_source.txt
Reason: Permission denied
test_exam1.c:166:test_copy_file_write_only_source:PASS
Successfully set MY_VAR to 42 using setenv()
Successfully set MY_VAR to 99 using setenv()
test_exam1.c:167:test_set_env_var:PASS
Retrieved environment variable: TEST_ENV: TestValue
NON_EXISTENT_ENV: Environment variable not found.
test_exam1.c:168:test_get_env_var:PASS

-----------------------
6 Tests 1 Failures 0 Ignored
FAIL
```

Conclusion

I learnt hard way to compile code using make tools.

Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
```

```
#include <fcntl.h>

#define BUFFER_SIZE 1024 // Define a constant for the buffer size

/**
 * Function: copy_file
 * --------------------
 * Copies the content from the source file to the destination file using system calls.
 *
 * Parameters:
 *    - source: The path of the source file (file to copy from).
 *    - destination: The path of the destination file (file to copy to).
 *
 * Write File Permissions: (Must use these file permissions)
 *  S_IRUSR: Owner has read permission.
 *  S_IWUSR: Owner has write permission.
 *  S_IRGRP: Group has read permission.
 *  S_IROTH: Others have read permission.
 *
 * Returns:
 *    0 on success
 *   -1 on failure (if an error occurs while opening, reading, or writing the files).
 *
 * System calls that MUST be used:
 *    - open(): Opens the source and destination files.
 *    - read(): Reads the content from the source file.
 *    - write(): Writes the content to the destination file.
 *    - close(): Closes the files after the operation.
 */
int copy_file(const char *source, const char *destination)
{

    // Check if file exists
    if (access(source, F_OK) != 0) {
        fprintf(stderr, "File does not exist: %s\n", source);
        perror("Reason");
        return -1;
    }

    // Check if file has read permission
    if (access(source, R_OK) != 0) {
        fprintf(stderr, "File does not have read permission: %s\n", source);
        perror("Reason");
        return -1;
    }

    //open source_file in readonly mode, it will return a valid file descriptor.
    // int open(const char *pathname, int flags);
    int read_fd = open(source, O_RDONLY);
    if (read_fd == -1)
    {
        fprintf(stderr, "Opening Error!: %s\n", source);
        perror("Reason");
        //perror("Error: when using opening source.");
        return -1;
    }
    else
    {
        printf("File: source Opened successfully.\n");
    }


    //open destination in 'write' mode.
    // "Open the file for writing. If it doesn’t exist, create it with permissions 664 (rw-rw-r--). If it already exists, truncate it.
    // int open(const char *pathname, int flags);
    int write_fd = open(destination, O_WRONLY | O_CREAT, S_IWUSR, O_TRUNC | S_IRUSR | S_IRGRP | S_IWGRP | S_IROTH );
    if (write_fd == -1)
    {
        perror("Error: when opening destination");
        return -1;
    }
    else
    {
        printf("File: destination Opened successfully\n");
    }


    //read 1024 bytes of data in one cycle
    char buffer[BUFFER_SIZE]; //already allocated a buffer of size 1024 bytes
    ssize_t bytes_read;

    //read will return > 0 as number of bytes it read, 0 as End of file, and -1 as error
    // it will read contents of file of size (buffer) into buffer.
    while ((bytes_read = read(read_fd, buffer, sizeof(buffer))) > 0) {
        if (write(write_fd, buffer, bytes_read) != bytes_read) {
            perror("Error writing to destination files");
            close(read_fd);
            close(write_fd);

            return -1;
        }
    }

    if (bytes_read == -1) {
        perror("Error reading from source file");
        return -1;
    } else {
        printf("File copied successfully to destination.\n");
    }

    // close
    close(read_fd);
    close(write_fd);
    return 0;

}

/**
 * Function: get_env_var
 * ----------------------
 * Retrieves the value of an environment variable.
 *
 * Parameters:
 *    - var_name: The name of the environment variable to retrieve.
 *
 * Returns:
 *    - The value of the environment variable if it exists.
 *    - A message "Environment variable not found" if the variable is not found.
 *
 * This function uses the standard library function getenv() to get the value.
 */
const char *get_env_var(const char *var_name)
{
    char *env_var_val = getenv(var_name);
    if (env_var_val) {
        printf("Retrieved environment variable: %s: %s\n", var_name, env_var_val);
        return env_var_val;
    } else {
        printf("%s: Environment variable not found.\n", var_name);
        return "Environment variable not found";
    }
}


/**
 * Function: set_env_var
 * ----------------------
 * Sets or updates the value of an environment variable.
 *
 * Parameters:
 *    - var_name: The name of the environment variable to set.
 *    - value: The value to assign to the environment variable.
 *    - overwrite: Determines whether to overwrite the variable if it already exists.
 *                 - 1 to overwrite, 0 to prevent overwriting.
 *
 * Returns:
 *    0 on success.
 *   -1 on failure (if setting the environment variable fails).
 */
int set_env_var(const char *var_name, const char *value, int overwrite)
{

    if (setenv(var_name, value, overwrite) != 0) {
        perror("Failed to set environment variable");
        return -1;
    } else {
        printf("Successfully set %s to %s using setenv()\n", var_name, getenv(var_name));
        return 0;
    }
}
```