

CSE 344
System Programming
4th Assignment
Report



Author

Mert Emir ŞEKER

200104004085

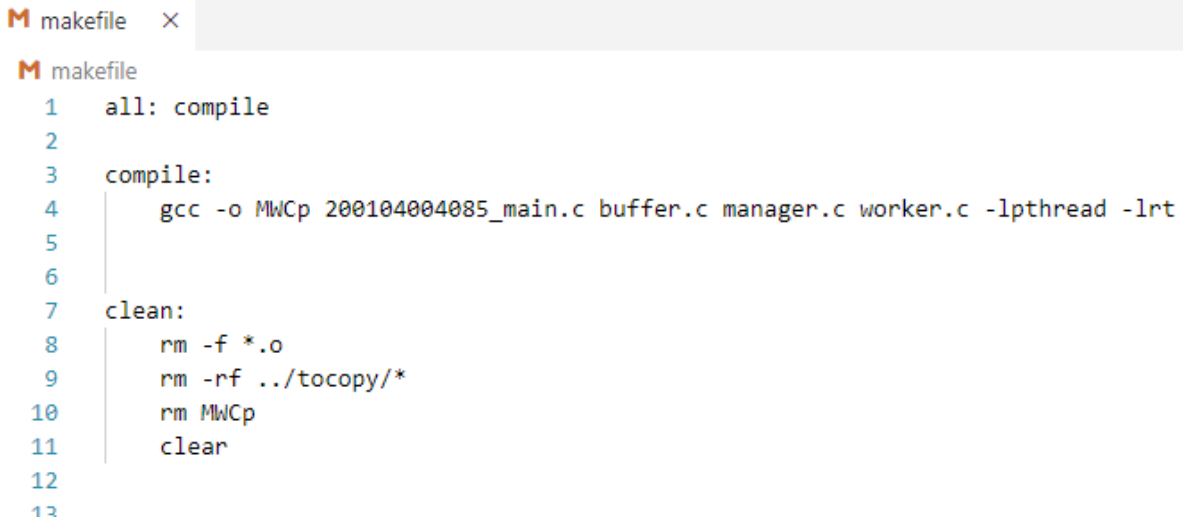
Date

26.05.2024

Table Of Contents

1. Makefile.....	3
2. Makefile Commands	3
3. How to run the code?	3
4. Code Explanation	4
4.1 Buffer.c and Buffer.h	4
4.2 Manager.c and Manager.h	6
4.3 Worker.c and Worker.h	8
4.4 Main.c	10
5. Example Outputs	13
5.1 Test 1 (in pdf)	13
5.2 Test 2 (in pdf)	13
5.3 Test 3 (in pdf)	14
5.4 ./MWCp 10 10 ../testdir ../tocopy	14
5.5 valgrind ./MWCp 1024 10 ../testdir ../tocopy	15
5.6 ./MWCp 1024 10 ../testdir ../tocopy	15
5.7 valgrind ./MWCp 10 4 ../testdir/src/libvterm/src ../newfile	16
5.8 valgrind ./MWCp 10 10 ../testdir ../toCopy	16
5.9 valgrind ./MWCp 10 10 ../testdir ../tocopy (with Ctrl-C signal)	17

1. Makefile:



```
M makefile ×
M makefile
1 all: compile
2
3 compile:
4 gcc -o MWCP 200104004085_main.c buffer.c manager.c worker.c -lpthread -lrt
5
6
7 clean:
8 rm -f *.o
9 rm -rf ../tocopy/*
10 rm MWCP
11 clear
12
13
```

2. Makefile Commands:

All: Compiles the code.

Compile: Compiles the code.

Clean: Clears terminal.

3. How to run the code?

To run the program, you can write **make all** or just **make** in terminal and then **./MWCP #buffer_size #ofworkers ../source_file_directory ../destination_file_directory**.

4. Code Explanation and pseudocodes:

4.1. **Buffer.c and Buffer.h:** A thread-safe buffer mechanism is implemented in this C code (buffer.c) and its accompanying header (buffer.h) in order to synchronize file descriptor operations between a management thread and many worker threads in a directory copying tool. The buffer, which is described in the buffer_str structure, leverages POSIX threading primitives for correct synchronization and stores file descriptors representing files that need to be duplicated. The buffer_init function allocates memory for file descriptors, initializes mutex and condition variables, specifies the size of the buffer, and initializes a number of indices and the done flag. By deleting synchronization primitives, freeing up memory, and shutting file descriptors, the buffer_destroy method tidys up. The management thread uses the buffer_add method to add file descriptors to the buffer, guaranteeing mutual exclusion and alerting worker threads when data becomes accessible. On the other hand, worker threads utilize buffer_remove to obtain file descriptors, guaranteeing mutual exclusion and alerting the manager when space becomes available. Buffer_set_done broadcasts to wake up any waiting threads and sets the done flag. The buffer.h header file includes function prototypes for buffer operations and signal handling, external variables for thread management and statistics tracking, and the essential data structures (buffer_str and file_descriptor). This method guarantees effective thread operation without idle waiting and appropriate resource cleaning afterward.

Pseudocode:

```
// Initialize the buffer
```

```
function buffer_init(buffer, size)
    set buffer size
    set buffer count to 0
    set buffer done flag to 0
    set buffer input index to 0
    set buffer output index to 0
    allocate memory for file descriptors
    if memory allocation fails
        print error message and exit
    initialize mutex
    initialize not full condition variable
    initialize not empty condition variable
// Destroy the buffer and free resources
function buffer_destroy(buffer)
    lock mutex
    for each file descriptor in the buffer
        close source file descriptor
        close destination file descriptor
    unlock mutex
    free file descriptor memory
    destroy mutex
```

```
    destroy not full condition variable
    destroy not empty condition variable
// Add a file descriptor to the buffer
function buffer_add(buffer, fd)
    lock mutex
    while buffer is full and not done
        wait for not full condition
    if done
        unlock mutex and return
    add file descriptor to buffer at input index
    update input index
    increment buffer count
    signal not empty condition
    unlock mutex

// Remove a file descriptor from the buffer
function buffer_remove(buffer)
    lock mutex
    while buffer is empty and not done
        wait for not empty condition
    if buffer is empty and done
        unlock mutex and return invalid file descriptor
    get file descriptor from buffer at output index
    update output index
    decrement buffer count
    signal not full condition
    unlock mutex
    return file descriptor

// Set the done flag of the buffer
function buffer_set_done(buffer)
    lock mutex
    set done flag
    broadcast not empty condition
    broadcast not full condition
    unlock mutex
```

4.2. **Manager.c and Manager.h:** The directory copying utility's manager thread functionality is implemented in this C file (manager.c) and the matching header (manager.h). In order for worker threads to handle file descriptors, the management thread must traverse the source directory, confirm that destination directories exist, and add file descriptors to the buffer. The `create_directory_if_not_exists` function determines if a directory already exists and, if not, creates it. By extracting the directory path and using `create_directory_if_not_exists`, the function `ensure_parent_directories_exists` verifies that all parent directories for a given file path exist. `Copy_file` is a function that copies files from one location to another, checks if parent directories are present, opens source and destination files, creates FIFOs if necessary, adds file descriptors to the buffer, and updates counters for normal and FIFO files. In order to process a directory, the function `process_directory` calls `copy_file` for files and `process_directory` recursively for subdirectories. It also verifies that the destination directory exists, opens the source directory, reads its entries, and processes each entry according to its type (directory, FIFO, or regular file). Initializing the required variables, determining if the source path is a directory, FIFO, or ordinary file, and processing it appropriately are all done by the manager function. It also constructs the root destination directory for a directory, runs `copy_file` for files, and sets the buffer's done flag when it's finished. The `manager.h` header file defines the function prototypes for buffer operations and signal handling, external variables for managing threads and tracking statistics, and the `manager_args` structure for passing arguments to the manager function. These definitions guarantee effective thread operation and appropriate resource cleanup after completion.

Pseudocode:

```
// Function to create a destination directory if it does not exist
function create_directory_if_not_exists(dir_path)
    if directory does not exist
        create directory
    if creation fails
        print error and exit

// Function to ensure the parent directories exist for the given file path
function ensure_parent_directories_exist(file_path)
    duplicate file path
    find the last slash in the path
    if last slash exists
        terminate string at the last slash
        create directory if it does not exist
    free duplicated string

// Function to copy a file from source to destination
function copy_file(src_path, dst_path, buffer, mode)
    ensure parent directories exist
    if file is FIFO
        create FIFO file
    if creation fails
        print error
```

```

        return
    increment FIFO file counter
else // Regular file
    open source file
    if opening fails
        print error
        return
    open destination file
    if opening fails
        print error
        close source file descriptor
        return
    create file descriptor structure
    add file descriptor to buffer
    increment regular file counter

// Function to process a directory
function process_directory(src_dir, dst_dir, buffer)
    ensure destination directory exists
    open source directory
    if opening fails
        print error
        set buffer as done
        return
    while reading directory entries and not done
        if entry is . or ..
            continue
        construct source path
        construct destination path
        get file status
        if getting status fails
            print error
            continue
        if entry is a directory
            recursively process directory
            increment directory counter
        else if entry is a FIFO
            copy FIFO file
        else if entry is a regular file
            copy regular file
    close source directory

// Manager function to handle copying process
function manager(arg)
    cast argument to manager_args pointer
    get buffer from arguments
    get source and destination paths from arguments
    get status of source path

```

```

if getting status fails
    print error
    set buffer as done
    return
if source path is a directory
    create root destination directory
    ensure root destination directory exists
    process source directory
else if source path is a FIFO
    construct destination file path
    copy FIFO file
else if source path is a regular file
    construct destination file path
    copy regular file
else
    print error for unsupported file type
    set buffer as done
    return
set buffer as done
return

```

5.3 Worker.c and Worker.h: The directory copying utility's worker thread functionality is implemented in this C code (worker.c) and the matching header (worker.h). Every worker thread updates the amount bytes transferred, transfers data from the source file to the destination file, and recovers file descriptors from the buffer. The worker thread begins by allocating memory for a buffer to read data from and converting the parameter to a `buffer_str` pointer. After that, the worker function goes into a loop where it keeps checking to see if the global done flag is set. If not, `buffer_remove` is used to extract a file descriptor pair (source and destination) from the buffer. The loop ends if an incorrect file descriptor—which indicates the termination condition—is retrieved. The function writes data in chunks to the destination file while it is in the loop, reading data from the source file into the buffer. It retries writing if a signal (EINTR) interrupts the write operation. When reading or writing errors occur, the file descriptors are closed and error warnings are printed. It produces a message indicating the completion of the copy operation and closes the source and destination file descriptors after copying. Thread-safe access to the shared variable is ensured by the worker thread locking a mutex to update the global total bytes copied (`total_bytes_copied`) upon leaving the loop. Ultimately, it returns and releases the buffer RAM that was allotted. The worker function prototype is defined in the header file `worker.h`, which also contains the headers required for file operations, memory management, and error handling.

Pseudocode:

```
// Worker thread function

function worker(arg)

    cast arg to buffer_str pointer

    allocate memory for buffer

    if memory allocation fails

        print error and return

    initialize total bytes copied by this thread to 0

    while not done

        get file descriptors from buffer

        if file descriptors are invalid

            break

        while read data from source file into buffer

            if read fails and is not interrupted by signal

                print error

                close file descriptors

                free buffer

                return

        write data from buffer to destination file

        while data remains to be written

            if write fails and is not interrupted by signal

                print error

                close file descriptors

                free buffer

                return

        update write pointer and bytes remaining

    update total bytes copied by this thread
```

```
if read fails and is not interrupted by signal
    print error

close file descriptors
print copy completion message

lock mutex
update global total bytes copied
unlock mutex

free buffer
return
```

5.4 Main.c: The main program in this C file initializes the buffer, creates the worker and manager threads, and handles signal interruptions in order to coordinate the directory copying function. To establish the buffer size, number of worker threads, source and destination directories, and other program parameters, the software first checks and parses the command-line options. It first configures a SIGINT signal handler to gracefully manage interruptions, and then it uses `gettimeofday` to calculate the operation's execution time. The application launches the manager thread, passes the source and destination directories, and initializes the buffer with the given size. Additionally, the buffer needed to process file descriptors is created for each of the designated number of worker threads. The done flag is set by the `handle_signal` function, which is designed to detect the SIGINT signal and signal the buffer to wake up any waiting threads. This signals the end of the copying operation. The manager and worker threads work simultaneously, with the workers carrying out the actual copying and the management adding file descriptors to the buffer. The program waits for each worker thread to accomplish its task once the management thread ends. After that, the buffer is destroyed, memory that has been allotted is released, and the overall execution time is calculated. Statistics regarding the operation are printed, including the number of files transferred, directories handled, total bytes copied, and the total time.

Pseudocode:

```
// Function to print usage message
function print_usage(prog_name)
    print "Usage: prog_name <buffer_size> <num_workers> <src_dir> <dst_dir>"

// Signal handler for SIGINT
function handle_signal(sig)
    if sig is SIGINT
        print "Ctrl-C signal received. Copying stopped."
        set interrupted and done flags to 1
        set buffer as done

// Function to print statistics
function print_statistics(num_workers, buffer_size, elapsed)
    calculate elapsed time in minutes, seconds, and milliseconds
    print statistics including number of files, directories, total bytes copied, and total time

// Main function
function main(argc, argv)
    if argc is not 5
        print_usage(argv[0])
        exit with failure

    parse buffer_size, num_workers, src_dir, dst_dir from argv
    if buffer_size or num_workers is less than or equal to 0
        print_usage(argv[0])
        exit with failure

    print welcome message

    set up signal handling for SIGINT

    create destination directory if it does not exist

    start measuring time

    initialize buffer with buffer_size

    print "Files are copying..."

    create and start manager thread with src_dir, dst_dir, and buffer

    allocate memory for worker threads
    if allocation fails
```

print error and exit with failure

create and start worker threads with buffer

wait for manager thread to finish

wait for all worker threads to finish

destroy buffer and free memory

stop measuring time and calculate elapsed time

print statistics with num_workers, buffer_size, and elapsed time

print "Files copy operation completed successfully. Goodbye!"

return 0

5. Example Outputs:

5.1. Test 1 (in pdf):

```
bash - put_your_codes_here - Visual Studio Code
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ make
gcc -o MWCP main.c buffer.c manager.c worker.c -lpthread -lrt
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ valgrind ./MWCP 10 10 ../testdir
../tocopy
==2908== Memcheck, a memory error detector
==2908== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==2908== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==2908== Command: ./MWCP 10 10 ../testdir ../tocopy
==2908==
Welcome to the MWCP directory copying system!!!
Files are copying...

----- STATISTICS -----
Consumers: 10 - Buffer Size: 10
Number of Regular Files: 3116
Number of FIFO Files: 0
Number of Directories: 151
TOTAL BYTES COPIED: 73520554
TOTAL TIME: 03:40.826 (min:sec.milli)
Files are copy operation completed successfully.
Goodbye!!!
==2908==
==2908== HEAP SUMMARY:
==2908==   in use at exit: 0 bytes in 0 blocks
==2908==   total heap usage: 3,292 allocs, 3,292 frees, 5,131,361 bytes allocated
==2908==
==2908== All heap blocks were freed -- no leaks are possible
==2908==
==2908== For counts of detected and suppressed errors, rerun with: -v
==2908== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$

bash - put_your_codes_here - Visual Studio Code
seker@seker-VirtualBox:~/Desktop/tocopy$ ls -l
total 0
seker@seker-VirtualBox:~/Desktop/tocopy$ ls -l
total 4
seker@seker-VirtualBox:~/Desktop/tocopy$ cd testdir
seker@seker-VirtualBox:~/Desktop/tocopy/testdir$ ls -l
total 124
-rw-rw-r-- 1 seker seker 695 May 26 15:14 appveyor.yml
-rw-rw-r-- 2 seker seker 4096 May 26 15:17 ci
-rw-rw-r-- 1 seker seker 182 May 26 15:14 configure
-rw-rw-r-- 1 seker seker 2907 May 26 15:16 CONTRIBUTING.md
-rw-rw-r-- 1 seker seker 23108 May 26 15:16 Filelist
-rw-rw-r-- 1 seker seker 21217 May 26 15:14 Makefile
drwx----- 3 seker seker 4096 May 26 15:14 nsis
drwx----- 2 seker seker 4096 May 26 15:14 pixmaps
drwx----- 2 seker seker 4096 May 26 15:14 README
-rw-rw-r-- 1 seker seker 5946 May 26 15:14 README.md
-rw-rw-r-- 1 seker seker 4900 May 26 15:14 README.txt
drwx----- 19 seker seker 4096 May 26 15:17 runtime
drwx----- 13 seker seker 12288 May 26 15:16 src
drwx----- 2 seker seker 4096 May 26 15:16 tools
-rw-rw-r-- 1 seker seker 3843 May 26 15:14 uninstall.txt
-rw-rw-r-- 1 seker seker 1812 May 26 15:17 vintutor.bat
-rw-rw-r-- 1 seker seker 2901 May 26 15:16 vintutor.com
seker@seker-VirtualBox:~/Desktop/tocopy/testdir$

bash - put_your_codes_here - Visual Studio Code
seker@seker-VirtualBox:~/Desktop/testdir$ ls -l
total 124
-rw-rw-r-- 1 seker seker 695 May 12 2019 appveyor.yml
drwxrwxr-x 2 seker seker 4096 May 25 17:48 ci
-rw-rw-r-- 1 seker seker 182 May 12 2019 configure
-rw-rw-r-- 1 seker seker 2907 May 12 2019 CONTRIBUTING.md
-rw-rw-r-- 1 seker seker 23108 May 12 2019 Filelist
-rw-rw-r-- 1 seker seker 21217 May 12 2019 Makefile
drwxrwxr-x 3 seker seker 4096 May 10 11:40 nsis
drwxrwxr-x 2 seker seker 4096 May 10 11:40 pixmaps
drwxrwxr-x 2 seker seker 4096 May 10 11:40 README
-rw-rw-r-- 1 seker seker 5946 May 12 2019 README.md
-rw-rw-r-- 1 seker seker 4900 May 12 2019 README.txt
drwxrwxr-x 19 seker seker 4096 May 10 11:31 runtime
drwxrwxr-x 15 seker seker 12288 May 10 11:40 src
drwxrwxr-x 2 seker seker 4096 May 10 11:40 tools
-rw-rw-r-- 1 seker seker 3843 May 12 2019 uninstall.txt
-rw-rw-r-- 1 seker seker 1812 May 12 2019 vintutor.bat
-rw-rw-r-- 1 seker seker 2901 May 12 2019 vintutor.com
seker@seker-VirtualBox:~/Desktop/testdir$
```

5.2. Test 2 (in pdf):

```
bash - Desktop - Visual Studio Code
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ make
gcc -o MWCP main.c buffer.c manager.c worker.c -lpthread -lrt
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ ./MWCP 10 4 ../testdir/src/libvterm
rm/src ../newfile
Welcome to the MWCP directory copying system!!!
Files are copying...

----- STATISTICS -----
Consumers: 4 - Buffer Size: 10
Number of Regular Files: 140
Number of FIFO Files: 0
Number of Directories: 2
TOTAL BYTES COPIED: 24873082
TOTAL TIME: 00:05.428 (min:sec.milli)
Files are copy operation completed successfully.
Goodbye!!!
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$

bash - Desktop - Visual Studio Code
seker@seker-VirtualBox:~/Desktop/testdir$ cd src/
seker@seker-VirtualBox:~/Desktop/testdir/src$ cd libvterm/
seker@seker-VirtualBox:~/Desktop/testdir/src/libvterm$ cd src
seker@seker-VirtualBox:~/Desktop/testdir/src/libvterm/src$ ls
autocmd.c      ex cmds2.c      gui photon.c    iid ole.c
beval.c        ex cmds2.c      gui w32 rc.h    infplist.xml
beval.h        ex cmds.h       gui x11.c       inexpand.c
bigvim4.bat    ex docmd.c      gui x11 pm.h    INSTALL
blob.c         ex eval.c       gui xmdlg.c     INSTALLLmi.txt
blowfish.c     ex getln.c      gui xmbw.c      INSTALLmac.txt
buffer.c       feature.h       gui xmbwp.h     installm.sh
change.c       fileio.c        gvim.exe.mmf   INSTALLpc.txt
channel.c      findfile.c      gvimExt        install-sh
charset.c      fold.c          gvimExt        INSTALLvms.txt
config.h.in    getchar.c      hangulim.c     INSTALLx.txt
config.mk.dist glib ime.cpp   hardcopy.c     iscopyty.h
config.mk.in   glib ime.h     globals.h      iscopyty.h
configure      gui at fs.c     if cscope.c    json.c
create cmdidxs.vim gui atna.c     if cscope.h    json.c
crypt.c        gui at sb.c     if lua.c       json test.c
crypt.zip.c    gui at sb.h     if mzsck.h     keyboard.c
debugger.c     gui beval.c     if mzsck.h     keymap.h
dehqp.py       gui b.c         if ole.cpp     keyword test.c
dict.c         gui dwrite.cpp  if ole.h       keymap.h
diff.c         gui dwrite.h    if ole.idl     MHT10.pdf
diagraph.c     gui gtk.c       if perl.msvc   parser.c
diml.idl       gui gtk f.c     if perl.fio.c  pen.c
dlldata.c     gui gtk h       if perl.xs     rect.h
dosinst.c     gui gtk res.xml if py both.h   state.c
dosinst.h     gui gtk vms.h   if python3.c   termScreen.c
edit.c        gui gtk x11.c   if python.c    unicode.c
encoding.c     gui h           if ruby.c       utf8.h
eval.c         gui h           if tcl.c        vterm.c
evalfunc.c     gui motif.c     if xcdrv.c     vterm internal.h
seker@seker-VirtualBox:~/Desktop/testdir/src/libvterm/src$
```

5.3. Test 3 (in pdf):

```
bash - Desktop - Visual Studio Code
# bash put_your_codes_here X
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ make
gcc -o MWCP main.c buffer.c manager.c worker.c -lpthread -lrt
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ ./MWCP 10 10 ../testdir ../tocopy
Welcome to the MWCP directory copying system!!
Files are copying...
----- STATISTICS -----
Consumers: 10 - Buffer Size: 10
Number of Regular Files: 3116
Number of FIFO Files: 0
Number of Directories: 151
TOTAL BYTES COPIED: 73520554
TOTAL TIME: 00:23.170 (min:sec.milli)
Files are copy operation completed successfully.
Goodbye!!!
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$

bash - Desktop - Visual Studio Code
# bash testdir X
seker@seker-VirtualBox:~/Desktop$ ls -l
total 28
drwxrwxr-x 2 seker seker 4096 May 13 20:07 hw3
drwxrwxr-x 3 seker seker 4096 May 26 15:07 midterm
drwxrwxr-x 3 seker seker 4096 May 26 15:52 put_your_codes_here
drwxrwxr-x 10 seker seker 4096 May 25 17:43 testdir
drwxrwxr-x 2 seker seker 4096 May 26 15:53 tocopy
seker@seker-VirtualBox:~/Desktop$ ls -l
total 24
drwxrwxr-x 2 seker seker 4096 May 13 20:07 hw3
drwxrwxr-x 3 seker seker 4096 May 4 23:06 midterm
drwxrwxr-x 3 seker seker 4096 May 26 15:53 put_your_codes_here
drwxrwxr-x 10 seker seker 4096 May 25 17:43 testdir
drwxrwxr-x 2 seker seker 4096 May 26 15:53 tocopy
drwxrwxr-x 3 seker seker 4096 May 26 15:53 tocopy
seker@seker-VirtualBox:~/Desktop$ cd tocopy/
seker@seker-VirtualBox:~/Desktop/tocopy$ ls -l
total 4
drwxrwxr-x 10 seker seker 4096 May 26 15:53 testdir
seker@seker-VirtualBox:~/Desktop/tocopy$ cd testdir/
seker@seker-VirtualBox:~/Desktop/tocopy/testdir$ ls -l
total 124
-rw-rw-r-- 1 seker seker 695 May 26 15:53 appveyor.yml
drwxrwxr-x 2 seker seker 4096 May 26 15:53 ci
-rw-rw-r-- 1 seker seker 182 May 26 15:53 configure
-rw-rw-r-- 1 seker seker 2907 May 26 15:53 CONTRIBUTING.md
-rw-rw-r-- 1 seker seker 23108 May 26 15:53 Filelist
-rw-rw-r-- 1 seker seker 21217 May 26 15:53 Makefile
drwxrwxr-x 3 seker seker 4096 May 26 15:53 nsis
drwxrwxr-x 2 seker seker 4096 May 26 15:53 pixmaps
drwxrwxr-x 2 seker seker 4096 May 26 15:53 README
-rw-rw-r-- 1 seker seker 5946 May 26 15:53 README.md
drwxrwxr-x 1 seker seker 4096 May 26 15:53 README.txt
drwxrwxr-x 10 seker seker 4096 May 26 15:53 runtime
drwxrwxr-x 15 seker seker 12288 May 26 15:53 src
drwxrwxr-x 2 seker seker 4096 May 26 15:53 tools
-rw-rw-r-- 1 seker seker 3843 May 26 15:53 uninstal.txt
-rw-rw-r-- 1 seker seker 1812 May 26 15:53 vintutor.bat
-rw-rw-r-- 1 seker seker 2901 May 26 15:53 vintutor.com
seker@seker-VirtualBox:~/Desktop/tocopy/testdir$
```

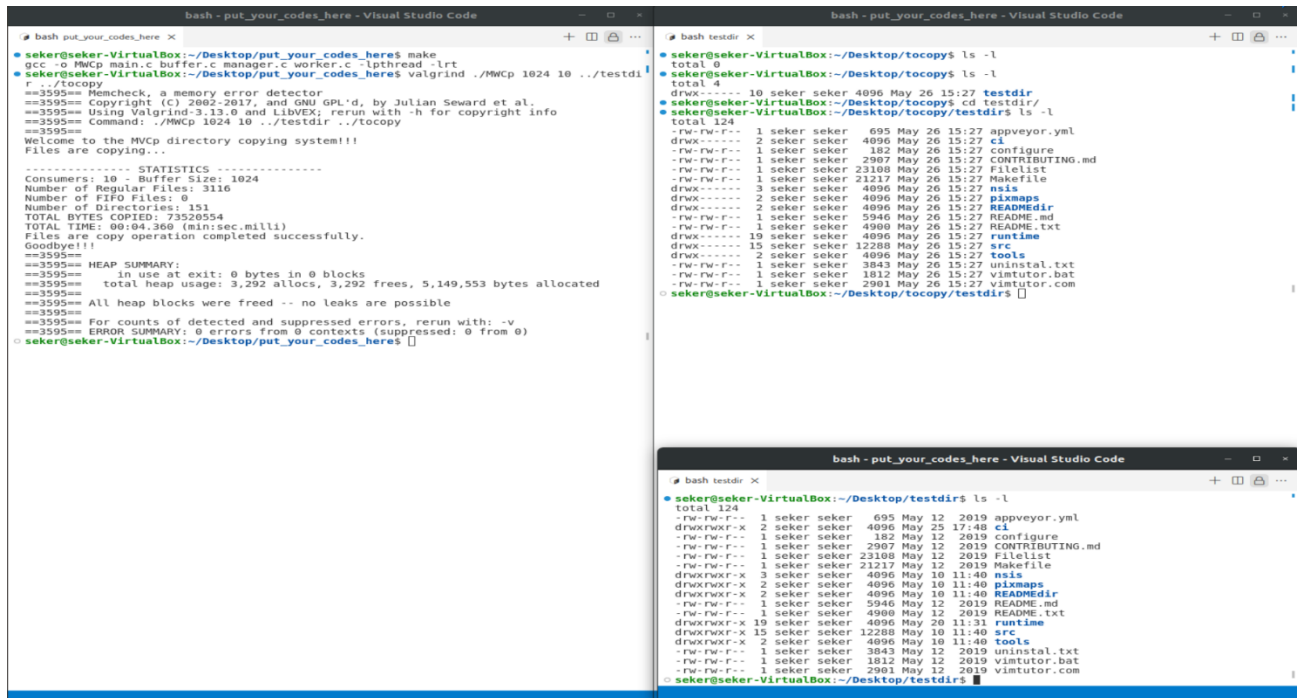
5.4. ./MWCP 10 10 ../testdir ../tocopy:

```
bash - put_your_codes_here - Visual Studio Code
# bash put_your_codes_here X
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ make
gcc -o MWCP main.c buffer.c manager.c worker.c -lpthread -lrt
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ ./MWCP 10 10 ../testdir ../tocopy
Welcome to the MWCP directory copying system!!
Files are copying...
----- STATISTICS -----
Consumers: 10 - Buffer Size: 10
Number of Regular Files: 3116
Number of FIFO Files: 0
Number of Directories: 151
TOTAL BYTES COPIED: 73520554
TOTAL TIME: 00:14.548 (min:sec.milli)
Files are copy operation completed successfully.
Goodbye!!!
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$

bash - put_your_codes_here - Visual Studio Code
# bash testdir X
seker@seker-VirtualBox:~/Desktop/tocopy$ ls -l
total 0
seker@seker-VirtualBox:~/Desktop/tocopy$ ls -l
total 4
drwxrwxr-x 10 seker seker 4096 May 26 15:23 testdir
seker@seker-VirtualBox:~/Desktop/tocopy/testdir$ ls
appveyor.yml  CONTRIBUTING.md  nsis  README.md  src  vintutor.bat
ci  Filelist  pixmaps  README.txt  tools  vintutor.com
configure  Makefile
seker@seker-VirtualBox:~/Desktop/tocopy/testdir$ ls -l
total 124
-rw-rw-r-- 1 seker seker 695 May 26 15:23 appveyor.yml
drwxrwxr-x 2 seker seker 4096 May 26 15:23 ci
-rw-rw-r-- 1 seker seker 182 May 26 15:23 configure
-rw-rw-r-- 1 seker seker 2907 May 26 15:23 CONTRIBUTING.md
-rw-rw-r-- 1 seker seker 23108 May 26 15:23 Filelist
-rw-rw-r-- 1 seker seker 21217 May 26 15:23 Makefile
drwxrwxr-x 3 seker seker 4096 May 26 15:23 nsis
drwxrwxr-x 2 seker seker 4096 May 26 15:23 pixmaps
drwxrwxr-x 2 seker seker 4096 May 26 15:23 README
-rw-rw-r-- 1 seker seker 5946 May 26 15:23 README.md
-rw-rw-r-- 1 seker seker 4909 May 26 15:23 README.txt
drwxrwxr-x 10 seker seker 4096 May 26 15:23 runtime
drwxrwxr-x 15 seker seker 12288 May 26 15:23 src
drwxrwxr-x 2 seker seker 4096 May 26 15:23 tools
-rw-rw-r-- 1 seker seker 3843 May 26 15:23 uninstal.txt
-rw-rw-r-- 1 seker seker 1812 May 26 15:23 vintutor.bat
-rw-rw-r-- 1 seker seker 2901 May 26 15:23 vintutor.com
seker@seker-VirtualBox:~/Desktop/tocopy/testdir$

bash - put_your_codes_here - Visual Studio Code
# bash testdir X
seker@seker-VirtualBox:~/Desktop/testdir$ ls -l
total 124
-rw-rw-r-- 1 seker seker 695 May 12 2019 appveyor.yml
drwxrwxr-x 2 seker seker 4096 May 25 17:48 ci
-rw-rw-r-- 1 seker seker 182 May 12 2019 configure
-rw-rw-r-- 1 seker seker 2907 May 12 2019 CONTRIBUTING.md
-rw-rw-r-- 1 seker seker 23108 May 12 2019 Filelist
-rw-rw-r-- 1 seker seker 21217 May 12 2019 Makefile
drwxrwxr-x 3 seker seker 4096 May 10 11:40 nsis
drwxrwxr-x 2 seker seker 4096 May 10 11:40 pixmaps
drwxrwxr-x 2 seker seker 4096 May 10 11:40 README
-rw-rw-r-- 1 seker seker 5946 May 12 2019 README.md
-rw-rw-r-- 1 seker seker 4909 May 12 2019 README.txt
drwxrwxr-x 10 seker seker 4096 May 20 11:31 runtime
drwxrwxr-x 15 seker seker 12288 May 10 11:40 src
drwxrwxr-x 2 seker seker 4096 May 10 11:40 tools
-rw-rw-r-- 1 seker seker 3843 May 12 2019 uninstal.txt
-rw-rw-r-- 1 seker seker 1812 May 12 2019 vintutor.bat
-rw-rw-r-- 1 seker seker 2901 May 12 2019 vintutor.com
seker@seker-VirtualBox:~/Desktop/testdir$
```

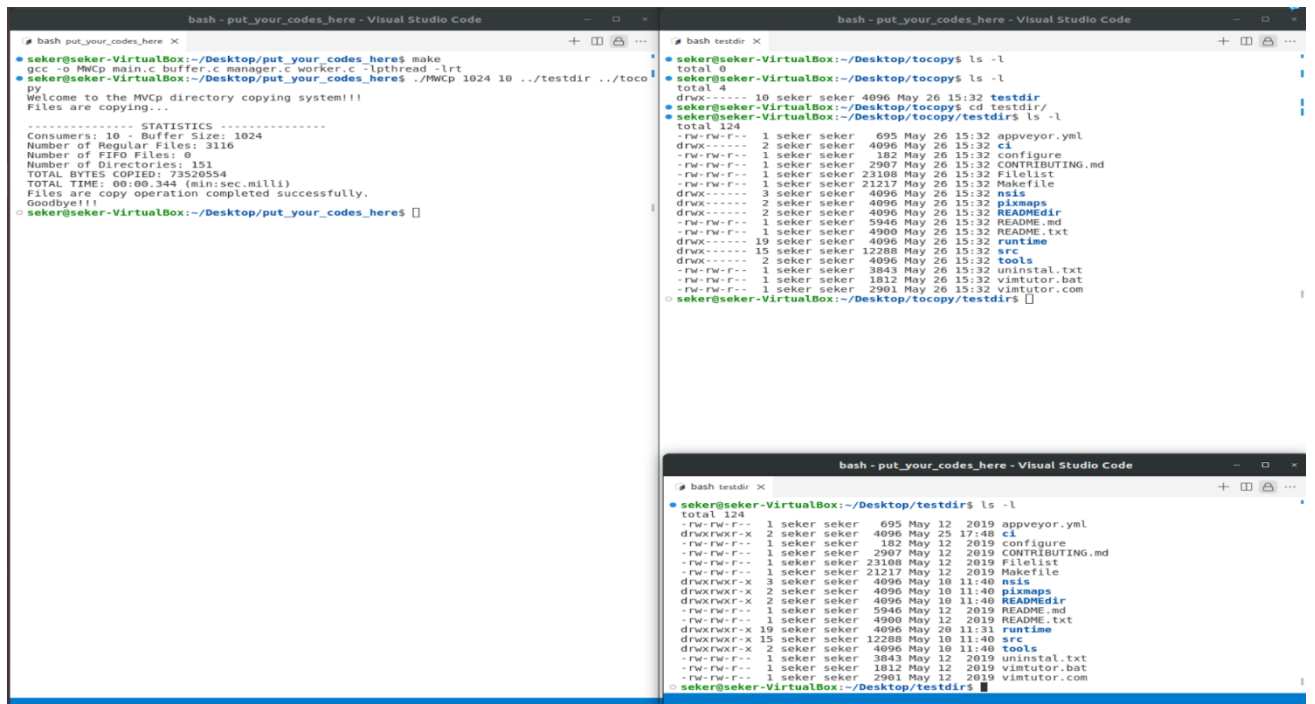
5.5. valgrind ./MWCp 1024 10 ../testdir ../tocopy:



```
# bash put_your_codes_here X
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ make
gcc -o MWCp main.c buffer.c manager.c worker.c -lpthread -lrt
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ valgrind ./MWCp 1024 10 ../testdir ../tocopy
==3595== Memcheck, a memory error detector
==3595== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==3595== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==3595== Command: ./MWCp 1024 10 ../testdir ../tocopy
==3595==
Welcome to the MWCp directory copying system!!!
Files are copying...
----- STATISTICS -----
Consumers: 10 - Buffer Size: 1024
Number of Regular Files: 3116
Number of FIFO Files: 0
Number of Directories: 151
TOTAL BYTES COPIED: 73520554
TOTAL TIME: 00:04.360 (min:sec.milli)
Files are copy operation completed successfully.
Goodbye!!!
==3595==
==3595== HEAP SUMMARY:
==3595==    in use at exit: 0 bytes in 0 blocks
==3595==    total heap usage: 3,292 allocs, 3,292 frees, 5,149,553 bytes allocated
==3595==
==3595== All heap blocks were freed -- no leaks are possible
==3595==
==3595== For counts of detected and suppressed errors, rerun with: -v
==3595== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$

# bash testdir X
seker@seker-VirtualBox:~/Desktop/tocopy$ ls -l
total 0
seker@seker-VirtualBox:~/Desktop/tocopy$ ls -l
total 4
drwx----- 10 seker seker 4096 May 26 15:27 testdir/
seker@seker-VirtualBox:~/Desktop/tocopy/testdir$ ls -l
total 124
-rw-rw-r-- 1 seker seker 695 May 26 15:27 appveyor.yml
drwx----- 2 seker seker 4096 May 26 15:27 ci
-rw-rw-r-- 1 seker seker 182 May 26 15:27 configure
-rw-rw-r-- 1 seker seker 2907 May 26 15:27 CONTRIBUTING.md
-rw-rw-r-- 1 seker seker 23108 May 26 15:27 Filelist
drwx----- 3 seker seker 4096 May 26 15:27 nsis
drwx----- 2 seker seker 4096 May 26 15:27 pixmaps
drwx----- 1 seker seker 5946 May 26 15:27 README.md
-rw-rw-r-- 1 seker seker 4900 May 26 15:27 README.txt
drwx----- 19 seker seker 4096 May 26 15:27 runtime
drwx----- 15 seker seker 12288 May 26 15:27 src
drwx----- 2 seker seker 4096 May 26 15:27 tools
-rw-rw-r-- 1 seker seker 3843 May 26 15:27 uninstall.txt
-rw-rw-r-- 1 seker seker 1812 May 26 15:27 vintutor.bat
-rw-rw-r-- 1 seker seker 2901 May 26 15:27 vintutor.com
seker@seker-VirtualBox:~/Desktop/tocopy/testdir$
```

5.6. ./MWCp 1024 10 ../testdir ../tocopy:



```
# bash put_your_codes_here X
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ make
gcc -o MWCp main.c buffer.c manager.c worker.c -lpthread -lrt
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ ./MWCp 1024 10 ../testdir ../tocopy
Welcome to the MWCp directory copying system!!!
Files are copying...
----- STATISTICS -----
Consumers: 10 - Buffer Size: 1024
Number of Regular Files: 3116
Number of FIFO Files: 0
Number of Directories: 151
TOTAL BYTES COPIED: 73520554
TOTAL TIME: 00:00.344 (min:sec.milli)
Files are copy operation completed successfully.
Goodbye!!!
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$

# bash testdir X
seker@seker-VirtualBox:~/Desktop/tocopy$ ls -l
total 0
seker@seker-VirtualBox:~/Desktop/tocopy$ ls -l
total 4
drwx----- 10 seker seker 4096 May 26 15:32 testdir/
seker@seker-VirtualBox:~/Desktop/tocopy/testdir$ ls -l
total 124
-rw-rw-r-- 1 seker seker 695 May 26 15:32 appveyor.yml
drwx----- 2 seker seker 4096 May 26 15:32 ci
-rw-rw-r-- 1 seker seker 182 May 26 15:32 configure
-rw-rw-r-- 1 seker seker 2907 May 26 15:32 CONTRIBUTING.md
-rw-rw-r-- 1 seker seker 23108 May 26 15:32 Filelist
drwx----- 3 seker seker 4096 May 26 15:32 nsis
drwx----- 2 seker seker 4096 May 26 15:32 pixmaps
drwx----- 1 seker seker 5946 May 26 15:32 README.md
-rw-rw-r-- 1 seker seker 4900 May 26 15:32 README.txt
drwx----- 19 seker seker 4096 May 26 15:32 runtime
drwx----- 15 seker seker 12288 May 26 15:32 src
drwx----- 2 seker seker 4096 May 26 15:32 tools
-rw-rw-r-- 1 seker seker 3843 May 26 15:32 uninstall.txt
-rw-rw-r-- 1 seker seker 1812 May 26 15:32 vintutor.bat
-rw-rw-r-- 1 seker seker 2901 May 26 15:32 vintutor.com
seker@seker-VirtualBox:~/Desktop/tocopy/testdir$

# bash testdir X
seker@seker-VirtualBox:~/Desktop/testdir$ ls -l
total 124
-rw-rw-r-- 1 seker seker 695 May 12 2019 appveyor.yml
drwxrwxr-x 2 seker seker 4096 May 25 17:48 ci
-rw-rw-r-- 1 seker seker 182 May 12 2019 configure
-rw-rw-r-- 1 seker seker 2907 May 12 2019 CONTRIBUTING.md
-rw-rw-r-- 1 seker seker 23108 May 12 2019 Filelist
drwxrwxr-x 3 seker seker 4096 May 10 11:40 nsis
drwxrwxr-x 2 seker seker 4096 May 10 11:40 pixmaps
drwxrwxr-x 2 seker seker 4096 May 10 11:40 README.md
-rw-rw-r-- 1 seker seker 5946 May 12 2019 README.txt
drwxrwxr-x 19 seker seker 4096 May 20 11:31 runtime
drwxrwxr-x 15 seker seker 12288 May 10 11:40 src
drwxrwxr-x 2 seker seker 4096 May 10 11:40 tools
-rw-rw-r-- 1 seker seker 3843 May 12 2019 uninstall.txt
-rw-rw-r-- 1 seker seker 1812 May 12 2019 vintutor.bat
-rw-rw-r-- 1 seker seker 2901 May 12 2019 vintutor.com
seker@seker-VirtualBox:~/Desktop/testdir$
```


5.7. valgrind ./MWCp 10 4 ../testdir/src/libvterm/src ../newfile:

```
# bash put_your_codes_here X
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ make
gcc -o MWCp main.c buffer.c manager.c worker.c -lpthread -lrt
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ valgrind ./MWCp 10 4 ../testdir/s
rc/libvterm/src ../newfile
==6515== Memcheck, a memory error detector
==6515== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==6515== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==6515== Command: ./MWCp 10 4 ../testdir/src/libvterm/src ../newfile
==6515==
Welcome to the MWCp directory copying system!!!
Files are copying...
----- STATISTICS -----
Consumers: 4 - Buffer Size: 10
Number of Regular Files: 140
Number of FIFO Files: 0
Number of Directories: 2
TOTAL BYTES COPIED: 24873062
TOTAL TIME: 00:29.679 (min:sec.milli)
Files are copy operation completed successfully.
Goodbye!!!
==6515==
==6515== HEAP SUMMARY:
==6515==   in use at exit: 0 bytes in 0 blocks
==6515==   total heap usage: 155 allocs, 155 frees, 104,758 bytes allocated
==6515==
==6515== All heap blocks were freed -- no leaks are possible
==6515==
==6515== For counts of detected and suppressed errors, rerun with: -v
==6515== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$

# bash src X
seker@seker-VirtualBox:~/Desktop/testdir/src/libvterm/src$ ls
autocmd.c      dosinst.h      gui.c          hashtable.c    installml.sh
beval.c        dosinst.h      gui dwrite.cpp if cscope.c    INSTALLLPC.txt
beval.h        edit.c         gui dwrite.h   if cscope.h    install.sh
bigvim4.bat    encoding.c     gui gtk.c      if lua.c       INSTALLVms.txt
bigvim.bat     eval.c         gui gtk f.c    if mzscl.h     INSTALLX.txt
blob.c         evalfunc.c     gui gtk f.h    if mzscl.h     iscgpty.c
blowfish.c     ex cmdids.h   gui gtk res.xml if ole.cpp     iscgpty.h
buffer.c        ex cmdid2.c   gui gtk vms.h  if ole.idl     lsl151090739430.pdf
change.c        ex cmdid.h    gui h          if perl_mvsc   json.c
channel.c       ex cmdid.h    gui h          if perl_mvsc   keymap.h
charset.c       ex docmd.c    gui mac.c      if perlfsio.c keyboard.c
config.h.in     ex eval.c     gui motif.c    if py both.h  keyword test.c
config.mk.dist  ex getln.c    gui photon.c   if python3.c  mouse.c
configure       feature.h     gui w32.c      if python3.c  mwt10.pdf
configure.ac    fileio.c      gui w32 rc.h   if ruby.c     parser.c
create cmdids.vim getchar.c     gui x11 pm.h   if tcl.c      pen.c
crypt.c         gtbl ime.cpp gui xmbw.c     if tcl.c      rect.h
debugger.c      gtbl ime.cpp gui xmbw.h     iid ole.c     state.c
dehgx.py        globas.h     gui xmbwp.h   intplist.xml  termstream.c
dict.c          gui at fs.c  gui xmbwp.h   inexpand.c    unicode.c
diff.c          gui at fs.c  gui xmbwp.h   inexpand.c    utf8.h
digraph.c       gui at fs.c  gui xmbwp.h   intplist.xml  vterm.c
diml.idl        gui at sb.h  hangulin.c    INSTALLL.txt  vterm internal.h
dlldata.c       gui beval.c  hardcopy.c    installman.sh
seker@seker-VirtualBox:~/Desktop/testdir/src/libvterm/src$

# bash src X
seker@seker-VirtualBox:~/Desktop/testdir/src/libvterm/src$ ls
autocmd.c      dosinst.h      gui.c          hashtable.c    installml.sh
beval.c        dosinst.h      gui dwrite.cpp if cscope.c    INSTALLLPC.txt
beval.h        edit.c         gui dwrite.h   if cscope.h    install.sh
bigvim4.bat    encoding.c     gui gtk.c      if lua.c       INSTALLVms.txt
bigvim.bat     eval.c         gui gtk f.c    if mzscl.h     INSTALLX.txt
blob.c         evalfunc.c     gui gtk f.h    if mzscl.h     iscgpty.c
blowfish.c     ex cmdids.h   gui gtk res.xml if ole.cpp     iscgpty.h
buffer.c        ex cmdid2.c   gui gtk vms.h  if ole.idl     lsl151090739430.pdf
change.c        ex cmdid.h    gui h          if perl_mvsc   json.c
channel.c       ex cmdid.h    gui h          if perl_mvsc   keymap.h
charset.c       ex docmd.c    gui mac.c      if perlfsio.c keyboard.c
config.h.in     ex eval.c     gui motif.c    if py both.h  keyword test.c
config.mk.dist  ex getln.c    gui photon.c   if python3.c  mouse.c
configure       feature.h     gui w32.c      if python3.c  mwt10.pdf
configure.ac    fileio.c      gui w32 rc.h   if ruby.c     parser.c
create cmdids.vim getchar.c     gui x11 pm.h   if tcl.c      pen.c
crypt.c         gtbl ime.cpp gui xmbw.c     if tcl.c      rect.h
debugger.c      gtbl ime.cpp gui xmbw.h     iid ole.c     state.c
dehgx.py        globas.h     gui xmbwp.h   intplist.xml  termstream.c
dict.c          gui at fs.c  gui xmbwp.h   inexpand.c    unicode.c
diff.c          gui at fs.c  gui xmbwp.h   inexpand.c    utf8.h
digraph.c       gui at fs.c  gui xmbwp.h   intplist.xml  vterm.c
diml.idl        gui at sb.h  hangulin.c    INSTALLL.txt  vterm internal.h
dlldata.c       gui beval.c  hardcopy.c    installman.sh
seker@seker-VirtualBox:~/Desktop/testdir/src/libvterm/src$
```

5.8. valgrind ./MWCp 10 10 ../testdir ../toCopy:

```
# bash put_your_codes_here X
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ make
gcc -o MWCp main.c buffer.c manager.c worker.c -lpthread -lrt
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ valgrind ./MWCp 10 10 ../testdir
../toCopy
==7143== Memcheck, a memory error detector
==7143== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==7143== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==7143== Command: ./MWCp 10 10 ../testdir ../toCopy
==7143==
Welcome to the MWCp directory copying system!!!
Files are copying...
----- STATISTICS -----
Consumers: 10 - Buffer Size: 10
Number of Regular Files: 3116
Number of FIFO Files: 0
Number of Directories: 151
TOTAL BYTES COPIED: 73520594
TOTAL TIME: 01:26.498 (min:sec.milli)
Files are copy operation completed successfully.
Goodbye!!!
==7143==
==7143== HEAP SUMMARY:
==7143==   in use at exit: 0 bytes in 0 blocks
==7143==   total heap usage: 3,292 allocs, 3,292 frees, 5,131,301 bytes allocated
==7143==
==7143== All heap blocks were freed -- no leaks are possible
==7143==
==7143== For counts of detected and suppressed errors, rerun with: -v
==7143== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
seker@seker-VirtualBox:~/Desktop/put_your_codes_here$

# bash testdir X
seker@seker-VirtualBox:~/Desktop/testdir$ ls -l
total 124
drwxrwxr-x 2 seker seker 4096 May 25 17:48 ci
drwxrwxr-x 1 seker seker 182 May 12 2019 appveyor.yml
drwxrwxr-x 1 seker seker 182 May 12 2019 configure
drwxrwxr-x 1 seker seker 2907 May 12 2019 CONTRIBUTING.md
drwxrwxr-x 1 seker seker 23108 May 12 2019 Filelist
drwxrwxr-x 1 seker seker 21217 May 12 2019 Makefile
drwxrwxr-x 3 seker seker 4096 May 10 11:40 nsis
drwxrwxr-x 2 seker seker 4096 May 10 11:40 pixmaps
drwxrwxr-x 2 seker seker 4096 May 10 11:40 README
drwxrwxr-x 1 seker seker 5946 May 12 2019 README.txt
drwxrwxr-x 19 seker seker 4096 May 26 11:31 runtime
drwxrwxr-x 15 seker seker 12288 May 10 11:40 src
drwxrwxr-x 2 seker seker 4096 May 10 11:40 tools
drwxrwxr-x 1 seker seker 3843 May 12 2019 uninstall.txt
drwxrwxr-x 1 seker seker 1812 May 12 2019 vimtutor.bat
drwxrwxr-x 1 seker seker 2901 May 12 2019 vimtutor.com
seker@seker-VirtualBox:~/Desktop/testdir$

# bash testdir X
seker@seker-VirtualBox:~/Desktop/testdir$ ls -l
total 124
drwxrwxr-x 2 seker seker 4096 May 25 17:48 ci
drwxrwxr-x 1 seker seker 182 May 12 2019 appveyor.yml
drwxrwxr-x 1 seker seker 182 May 12 2019 configure
drwxrwxr-x 1 seker seker 2907 May 12 2019 CONTRIBUTING.md
drwxrwxr-x 1 seker seker 23108 May 12 2019 Filelist
drwxrwxr-x 1 seker seker 21217 May 12 2019 Makefile
drwxrwxr-x 3 seker seker 4096 May 10 11:40 nsis
drwxrwxr-x 2 seker seker 4096 May 10 11:40 pixmaps
drwxrwxr-x 2 seker seker 4096 May 10 11:40 README
drwxrwxr-x 1 seker seker 5946 May 12 2019 README.txt
drwxrwxr-x 19 seker seker 4096 May 26 11:31 runtime
drwxrwxr-x 15 seker seker 12288 May 10 11:40 src
drwxrwxr-x 2 seker seker 4096 May 10 11:40 tools
drwxrwxr-x 1 seker seker 3843 May 12 2019 uninstall.txt
drwxrwxr-x 1 seker seker 1812 May 12 2019 vimtutor.bat
drwxrwxr-x 1 seker seker 2901 May 12 2019 vimtutor.com
seker@seker-VirtualBox:~/Desktop/testdir$
```


5.9. valgrind ./MWCp 10 10 ../testdir ../tocopy (with Ctrl-C signal):

```
bash - Desktop - Visual Studio Code
# bash put_your_codes_here X
• seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ make
gcc -o MWCp main.c buffer.c manager.c worker.c -lpthread -lrt
• seker@seker-VirtualBox:~/Desktop/put_your_codes_here$ valgrind ./MWCp 10 10 ../testdir
../tocopy
==7887== Memcheck, a memory error detector
==7887== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==7887== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==7887== Command: ./MWCp 10 10 ../testdir ../tocopy
==7887==
Welcome to the MWCp directory copying system!!!
Files are copying...
^C Ctrl-C signal received.
Copying stopped.

----- STATISTICS -----
Consumers: 10 - Buffer Size: 10
Number of Regular Files: 585
Number of FIFO Files: 0
Number of Directories: 23
TOTAL BYTES COPIED: 5703886
TOTAL TIME: 00:07.174 (min:sec.milli)
Files are copy operation completed successfully.
Goodbye!!!
==7887==
==7887== HEAP SUMMARY:
==7887==    in use at exit: 0 bytes in 0 blocks
==7887==    total heap usage: 633 allocs, 633 frees, 819,569 bytes allocated
==7887==
==7887== All heap blocks were freed -- no leaks are possible
==7887==
==7887== For counts of detected and suppressed errors, rerun with: -v
==7887== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
• seker@seker-VirtualBox:~/Desktop/put_your_codes_here$

bash - Desktop - Visual Studio Code
• seker@seker-VirtualBox:~/Desktop$ cd tocopy/
• seker@seker-VirtualBox:~/Desktop/tocopy$ ls
total 0
• seker@seker-VirtualBox:~/Desktop/tocopy$ ls -l
total 4
drwx----- 6 seker seker 4096 May 26 15:59 testdir
• seker@seker-VirtualBox:~/Desktop/tocopy$ cd testdir/
• seker@seker-VirtualBox:~/Desktop/tocopy/testdir$ ls -l
total 64
-rw-rw-r-- 1 seker seker 695 May 26 15:59 appveyor.yml
-rw-rw-r-- 1 seker seker 182 May 26 15:59 configure
-rw-rw-r-- 1 seker seker 21217 May 26 15:59 Makefile
drwx----- 3 seker seker 4096 May 26 15:59 nsis
drwx----- 2 seker seker 4096 May 26 15:59 READMEdir
-rw-rw-r-- 1 seker seker 5946 May 26 15:59 README.md
-rw-rw-r-- 1 seker seker 4908 May 26 15:59 README.txt
drwx----- 6 seker seker 4096 May 26 15:59 src
-rw-rw-r-- 1 seker seker 3843 May 26 15:59 uninstal.txt
• seker@seker-VirtualBox:~/Desktop/tocopy/testdir$

bash - Desktop - Visual Studio Code
• bash testdir X
• seker@seker-VirtualBox:~/Desktop/testdir$ ls -l
total 124
-rw-rw-r-- 1 seker seker 695 May 12 2019 appveyor.yml
drwxrwxr-x 2 seker seker 4096 May 25 17:48 ci
-rw-rw-r-- 1 seker seker 182 May 12 2019 configure
-rw-rw-r-- 1 seker seker 2907 May 12 2019 CONTRIBUTING.md
-rw-rw-r-- 1 seker seker 23168 May 12 2019 Filelist
-rw-rw-r-- 1 seker seker 21217 May 12 2019 Makefile
drwxrwxr-x 3 seker seker 4096 May 10 11:40 nsis
drwxrwxr-x 2 seker seker 4096 May 10 11:40 pixmaps
drwxrwxr-x 2 seker seker 4096 May 10 11:40 READMEdir
-rw-rw-r-- 1 seker seker 5946 May 12 2019 README.md
-rw-rw-r-- 1 seker seker 4908 May 12 2019 README.txt
drwxrwxr-x 19 seker seker 4096 May 20 11:31 runtime
drwxrwxr-x 15 seker seker 12288 May 10 11:40 src
drwxrwxr-x 2 seker seker 4096 May 10 11:40 tools
-rw-rw-r-- 1 seker seker 3843 May 12 2019 uninstal.txt
-rw-rw-r-- 1 seker seker 1812 May 12 2019 vintutor.bat
-rw-rw-r-- 1 seker seker 2901 May 12 2019 vintutor.com
• seker@seker-VirtualBox:~/Desktop/testdir$
```