CSE 344

HW5

Burak Kocausta

1901042605

Content

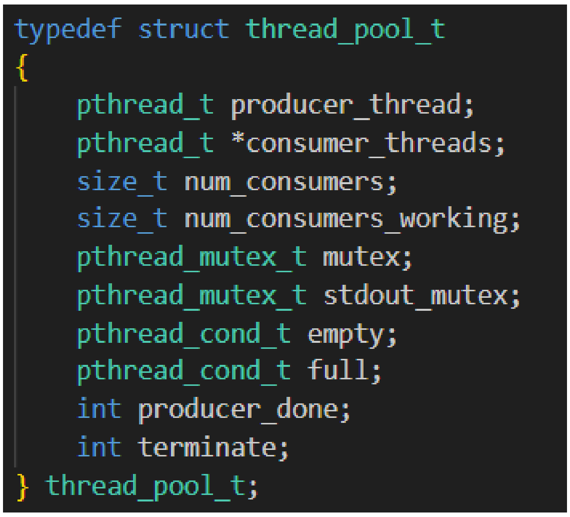
1- System Architecture and Problem Solution Approach

- I created a thread pool structure initially, and this structure holds consumer threads and producer thread. Using this structure I implemented 2 different worker threads, one for producer and one for consumers. Main program gets the inputs from command line it includes buffer size, number of consumer threads, source, and destination directory. Then it initializes the threads, and they start. It is a producer bounded thread, so it always starts with producing, and situation is decided with "producer_done" flag. The producer is done when all files are copied or there is a sigint signal. Consumers terminate when there are no items in the buffer. I synchronized them using two mutex, and three conditional variables. There are simple mutex for accessing the critical region, other mutex is for printing stdout. Conditional variables are empty, and full. When buffer is full, producer waits, and consumers consume all the buffer, when buffer is empty then producer becomes active. There are other cases except empty, full situation; one of them is maximum number of open file descriptors, if producer reached the maximum number of open file descriptor per process limit, it waits producer to do its job without exceeding this limit. Producer traverses the source directory recursively, and creates every directory, every file inside it, then passes the descriptors to the buffer. I represent this buffer with file name (path), and read file descriptor, write file descriptor. Consumer copies files with a limit of 4096 byte buffer because of not exceeding the memory limit. If there is a file which has size 4 GiB it is copied 4096 bytes by 4096 bytes. If the size of the buffer increases, copying will be faster. My task pool is a simple queue. I used queue data structure because I just poll task from there, and offer task both operations are constant time operations. I defined a structure for checking statistics, It includes number of directories copied, number of regular files copied, number of FIFOs copied. Also, total byte transferred is checked. I printed the total time in microseconds. “gettimeofday” function is used for that. It is the time between initializing the first thread and joining the last thread.

2 – Implementation Details and Requirements Met

Thread pool:

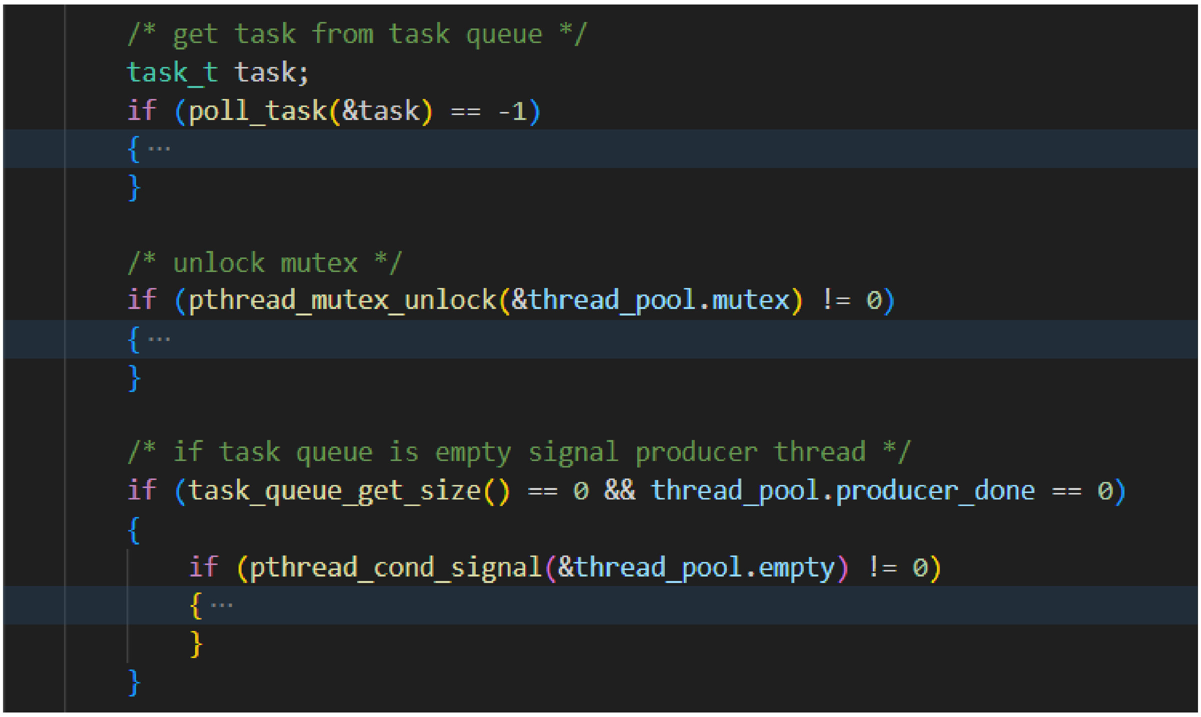
- Thread pool is implemented, it includes consumer threads, and producer thread.



```c
typedef struct thread_pool_t
{
    pthread_t producer_thread;
    pthread_t *consumer_threads;
    size_t num_consumers;
    size_t num_consumers_working;
    pthread_mutex_t mutex;
    pthread_mutex_t stdout_mutex;
    pthread_cond_t empty;
    pthread_cond_t full;
    int producer_done;
    int terminate;
} thread_pool_t;
```

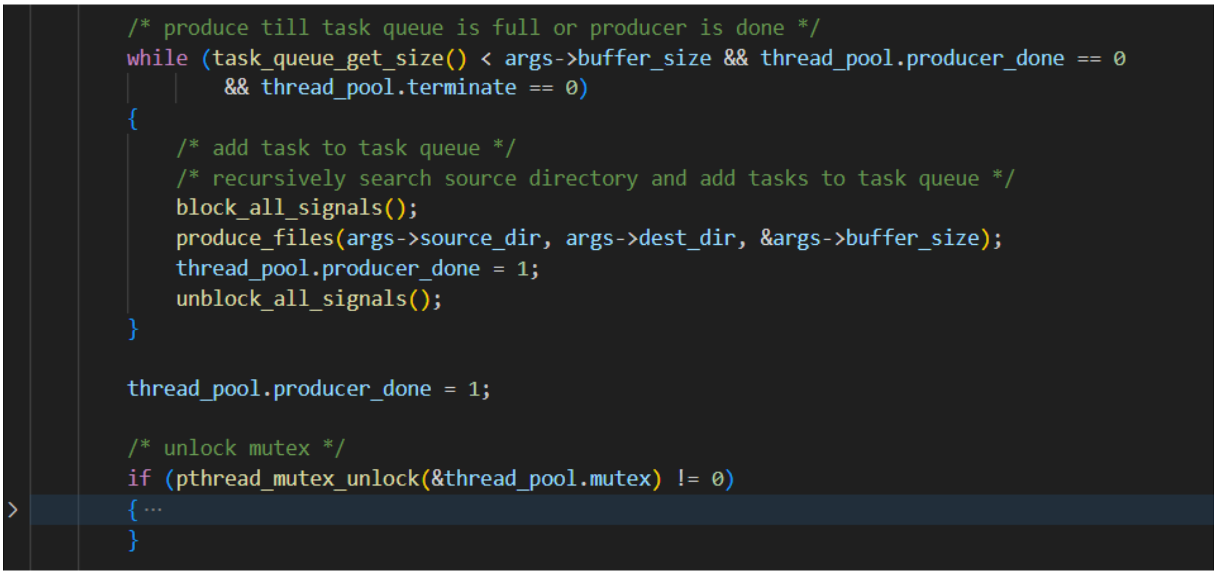
Synchronization and Preventing Data Race/Corruption:

- Synchronization is provided with mutexes and condition variables.
- Example synchronization situations for consumer thread.



```
/* get task from task queue */
task_t task;
if (poll_task(&task) == -1)
{…
}

/* unlock mutex */
if (pthread_mutex_unlock(&thread_pool.mutex) != 0)
{…
}

/* if task queue is empty signal producer thread */
if (task_queue_get_size() == 0 && thread_pool.producer_done == 0)
{
    if (pthread_cond_signal(&thread_pool.empty) != 0)
    {…
    }
}
```

- Mutex is unlocked when task is polled in consumer thread, and producer thread.

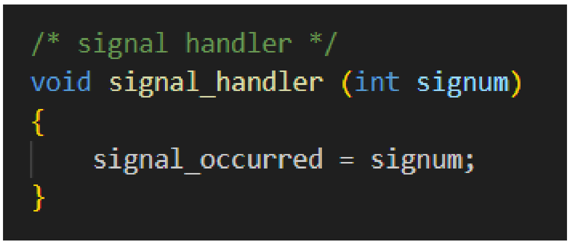


```
/* produce till task queue is full or producer is done */
while (task_queue_get_size() < args->buffer_size && thread_pool.producer_done == 0
        && thread_pool.terminate == 0)
{
    /* add task to task queue */
    /* recursively search source directory and add tasks to task queue */
    block_all_signals();
    produce_files(args->source_dir, args->dest_dir, &args->buffer_size);
    thread_pool.producer_done = 1;
    unblock_all_signals();
}

thread_pool.producer_done = 1;

/* unlock mutex */
if (pthread_mutex_unlock(&thread_pool.mutex) != 0)
{…
}
```

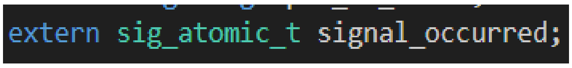
- Mutex is unlocked when producing is done, and done flag is set.

Signal Handling:

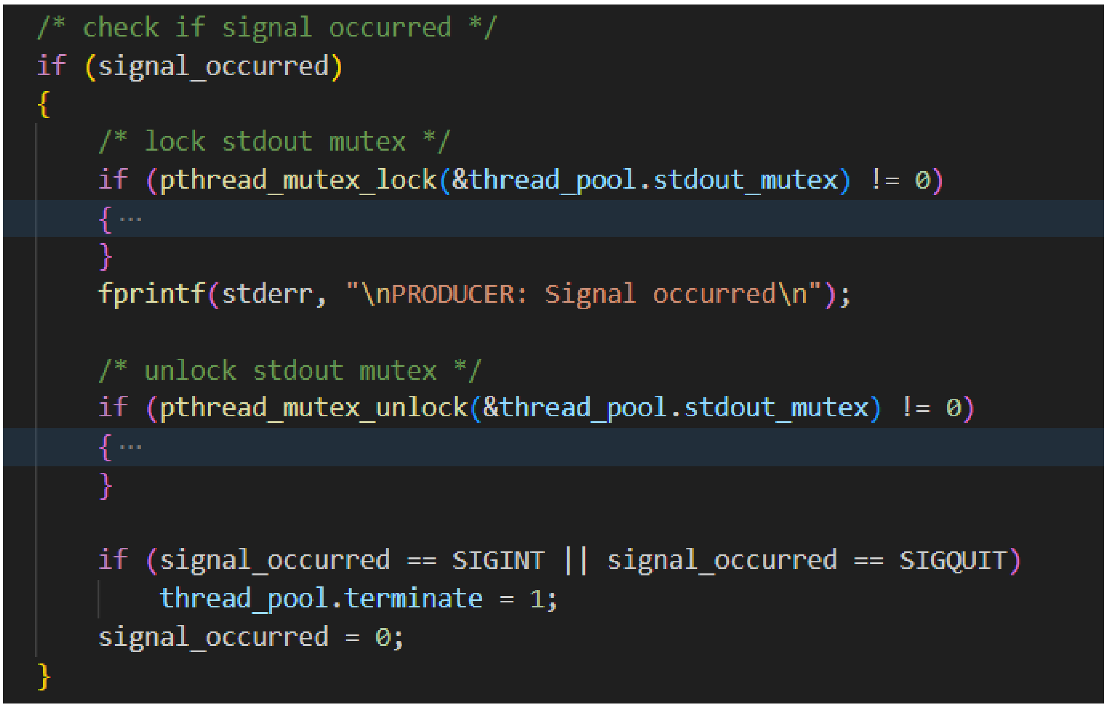
- One signal variable is used for signal checking. Each thread checks for signal, if it is a termination signal like SIGINT consumer finishes the buffer and terminates. Producer done producing.



```
/* signal handler */
void signal_handler (int signum)
{
    signal_occurred = signum;
}
```

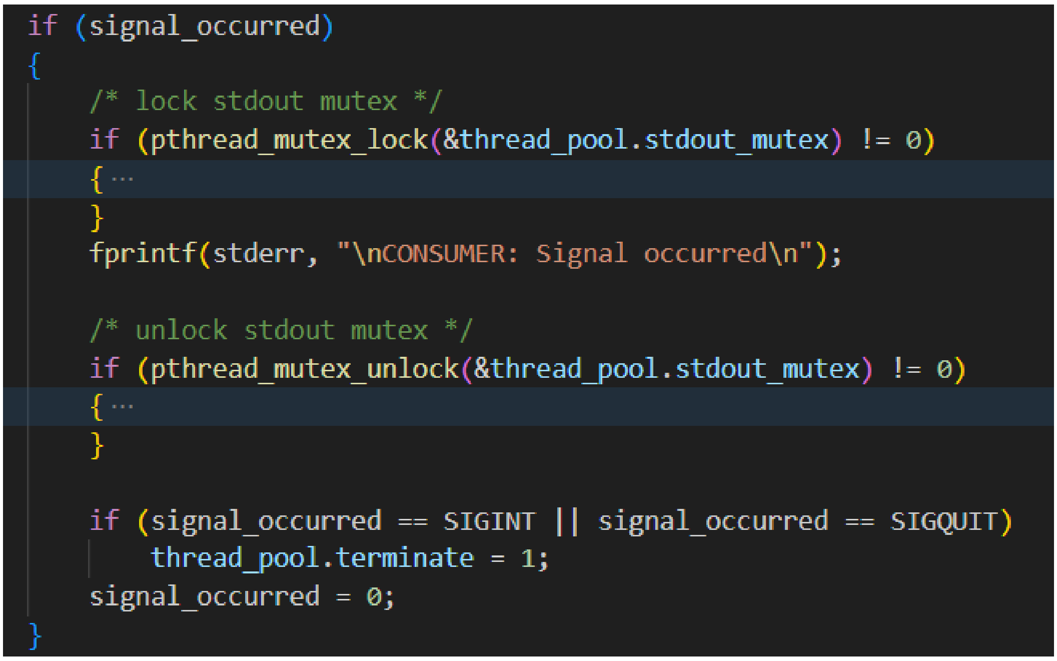


```
extern sig_atomic_t signal_occurred;
```



```
/* check if signal occurred */
if (signal_occurred)
{
    /* lock stdout mutex */
    if (pthread_mutex_lock(&thread_pool.stdout_mutex) != 0)
    {…
    }
    fprintf(stderr, "\nPRODUCER: Signal occurred\n");

    /* unlock stdout mutex */
    if (pthread_mutex_unlock(&thread_pool.stdout_mutex) != 0)
    {…
    }

    if (signal_occurred == SIGINT || signal_occurred == SIGQUIT)
        thread_pool.terminate = 1;
    signal_occurred = 0;
}
```

- Producer sets termination flag.

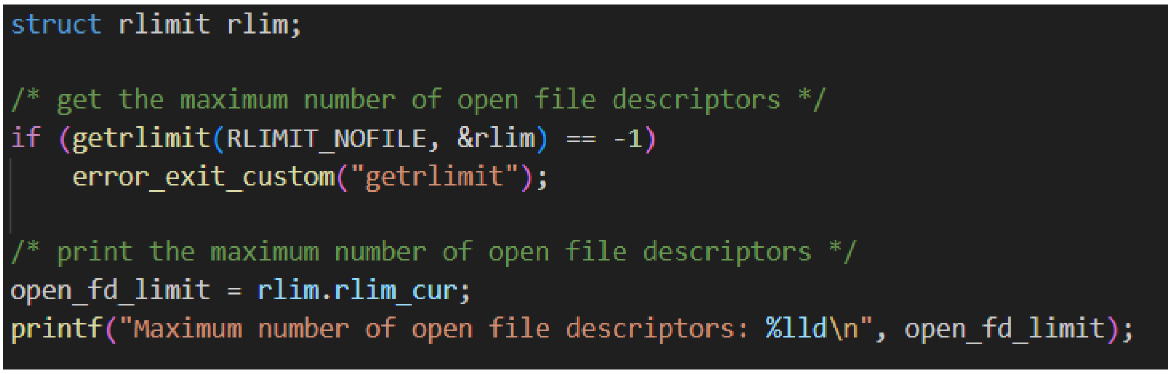


```
if (signal_occurred)
{
    /* lock stdout mutex */
    if (pthread_mutex_lock(&thread_pool.stdout_mutex) != 0)
    {…
    }
    fprintf(stderr, "\nCONSUMER: Signal occurred\n");

    /* unlock stdout mutex */
    if (pthread_mutex_unlock(&thread_pool.stdout_mutex) != 0)
    {…
    }

    if (signal_occurred == SIGINT || signal_occurred == SIGQUIT)
        thread_pool.terminate = 1;
    signal_occurred = 0;
}
```

- Consumer sets termination flag.

Maximum Number of Open File Descriptors:

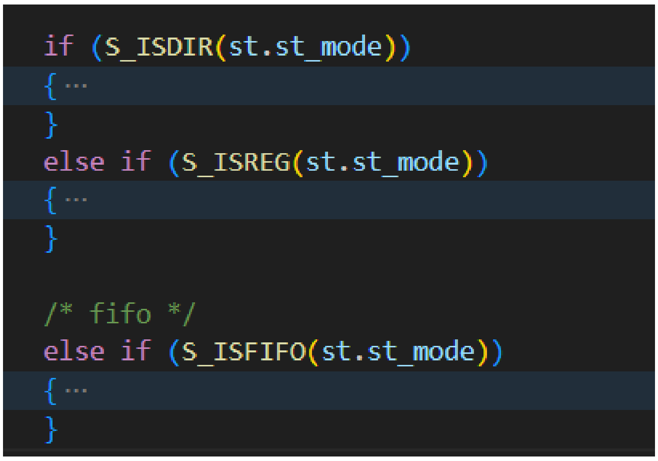
- I checked this situation using “getrlimit” function with “RLIMIT_NOFILE”. I set a global variable for this value, and producer thread checks this value before opening a file. If limit is reached, it waits for consumers to consume.



```
struct rlimit rlim;

/* get the maximum number of open file descriptors */
if (getrlimit(RLIMIT_NOFILE, &rlim) == -1)
    error_exit_custom("getrlimit");

/* print the maximum number of open file descriptors */
open_fd_limit = rlim.rlim_cur;
printf("Maximum number of open file descriptors: %lld\n", open_fd_limit);
```

File Types:

- All directory, regular file, and FIFOs are handled.



```
  if (S_ISDIR(st.st_mode))
  { ···
  }
  else if (S_ISREG(st.st_mode))
  { ···
  }

  /* fifo */
  else if (S_ISFIFO(st.st_mode))
  { ···
  }
```

Memory Handling:

- I used heap memory for task buffer, and copy buffer. Both are handled. All allocated memory is freed.

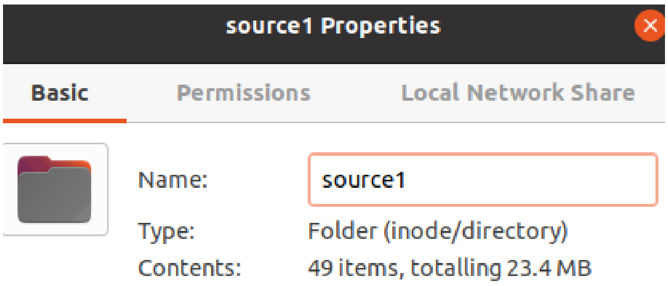
3 – Test Results

Input format:

./main <buffer size> <number of consumers> <sourcedir> <destdir>

TEST 1:

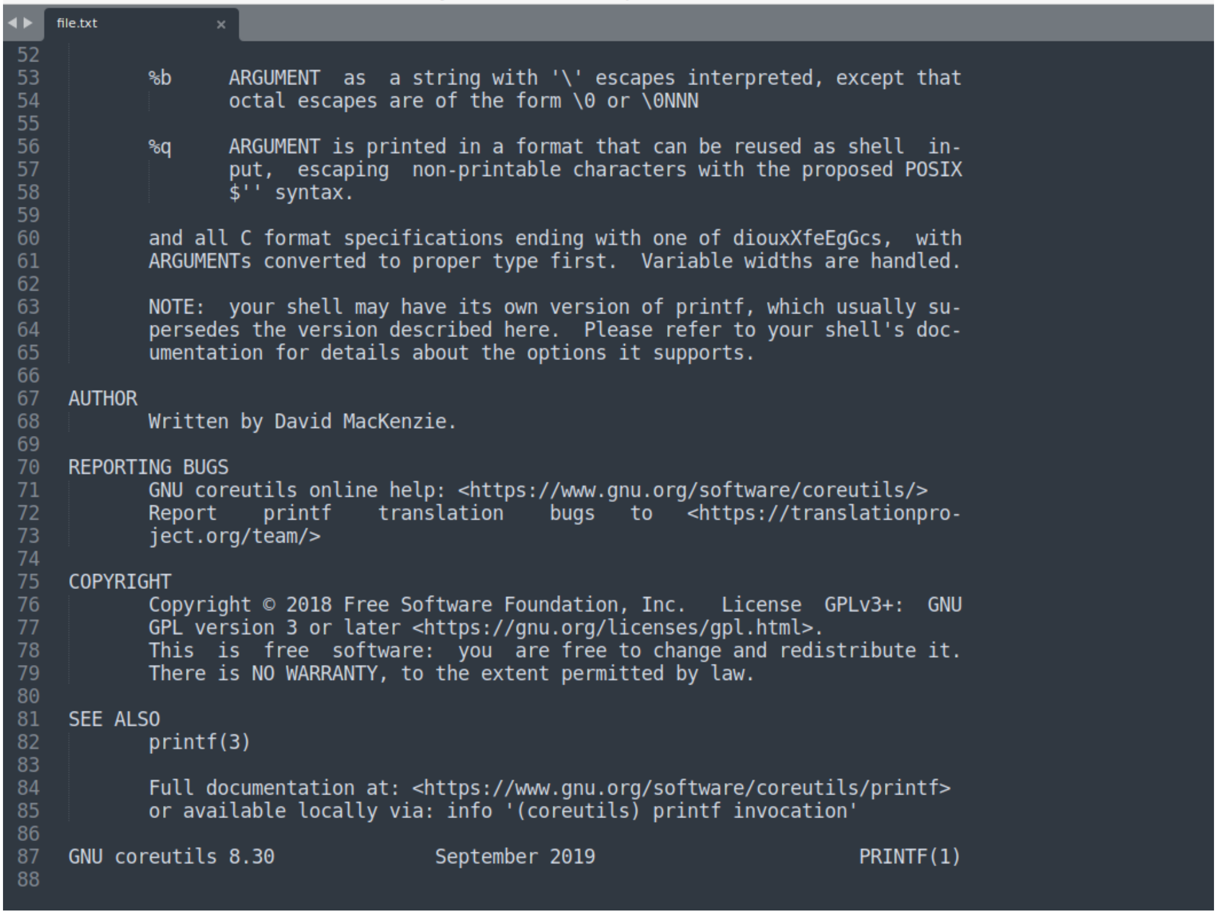
- Copy from source1 to dest1. File size is 23.4 MB.



Name: source1

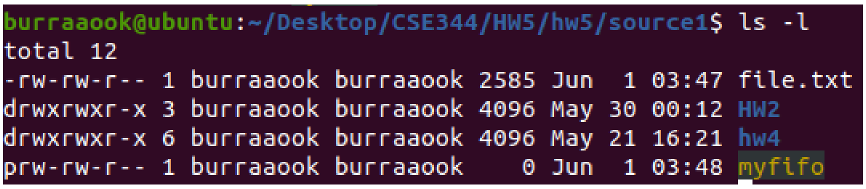
Type: Folder (inode/directory)

Contents: 49 items, totalling 23.4 MB

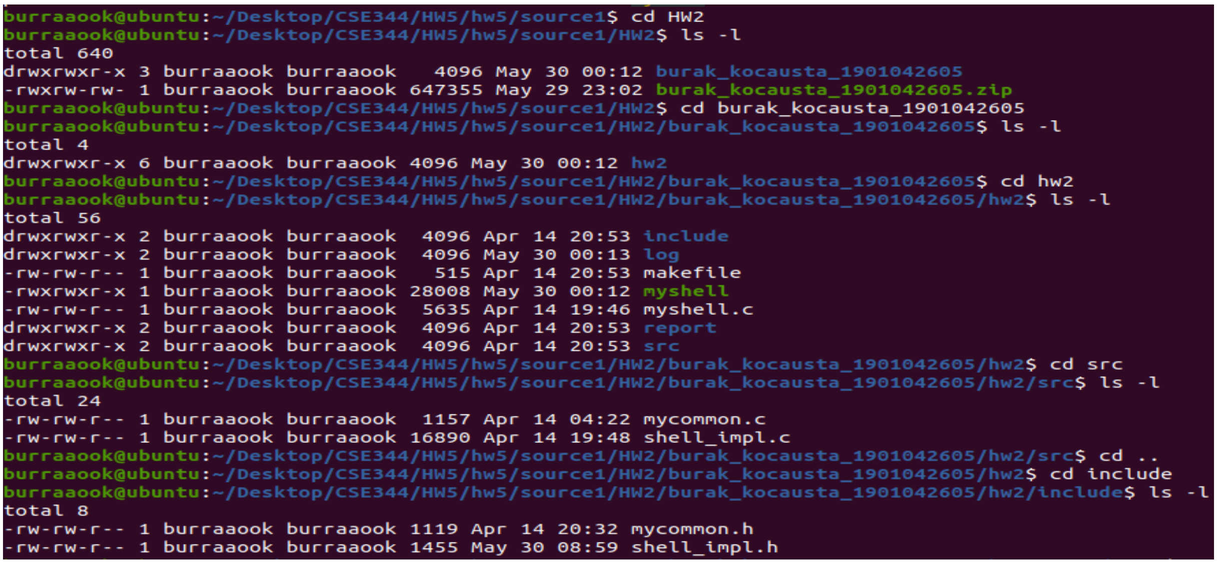


```

          %b     ARGUMENT  as  a string with '\' escapes interpreted, except that
                 octal escapes are of the form \0 or \0NNN

          %q     ARGUMENT is printed in a format that can be reused as shell  in-
                 put,  escaping  non-printable characters with the proposed POSIX
                 $'' syntax.

          and all C format specifications ending with one of diouxXfeEgGcs,  with
          ARGUMENTs converted to proper type first.  Variable widths are handled.

          NOTE:  your shell may have its own version of printf, which usually su-
          persedes the version described here.  Please refer to your shell's doc-
          umentation for details about the options it supports.

AUTHOR
          Written by David MacKenzie.

REPORTING BUGS
          GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
          Report    printf    translation    bugs    to    <https://translationpro-
          ject.org/team/>

COPYRIGHT
          Copyright © 2018 Free Software Foundation, Inc.   License  GPLv3+:  GNU
          GPL version 3 or later <https://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.

SEE ALSO
          printf(3)

          Full documentation at: <https://www.gnu.org/software/coreutils/printf>
          or available locally via: info '(coreutils) printf invocation'

GNU coreutils 8.30             September 2019                        PRINTF(1)

```

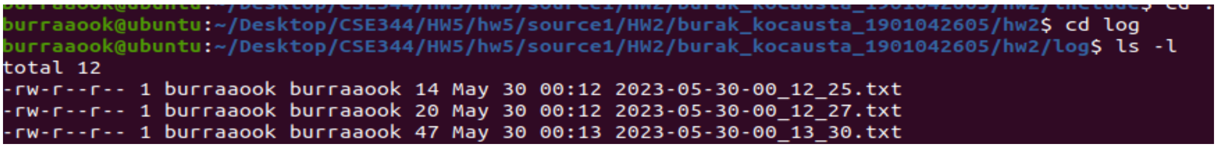
- Manual of printf



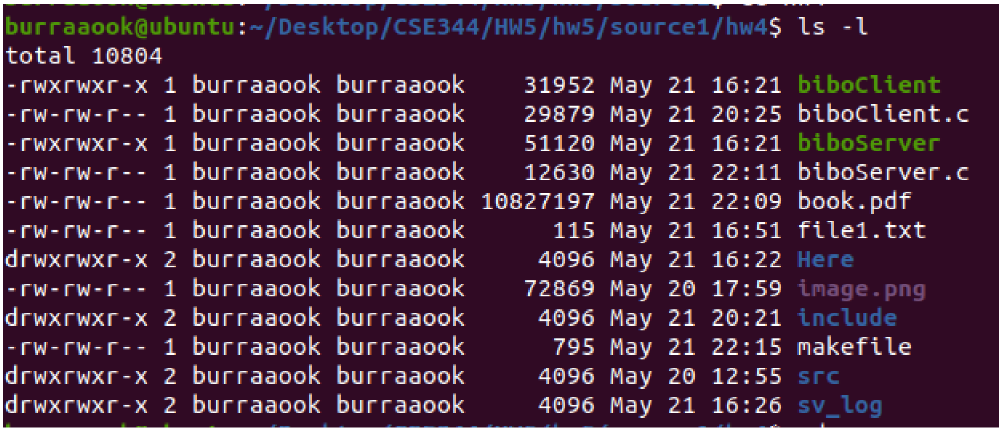
```
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1$ ls -l
total 12
-rw-rw-r-- 1 burraaook burraaook 2585 Jun  1 03:47 file.txt
drwxrwxr-x 3 burraaook burraaook 4096 May 30 00:12 HW2
drwxrwxr-x 6 burraaook burraaook 4096 May 21 16:21 hw4
prw-rw-r-- 1 burraaook burraaook    0 Jun  1 03:48 myfifo
```



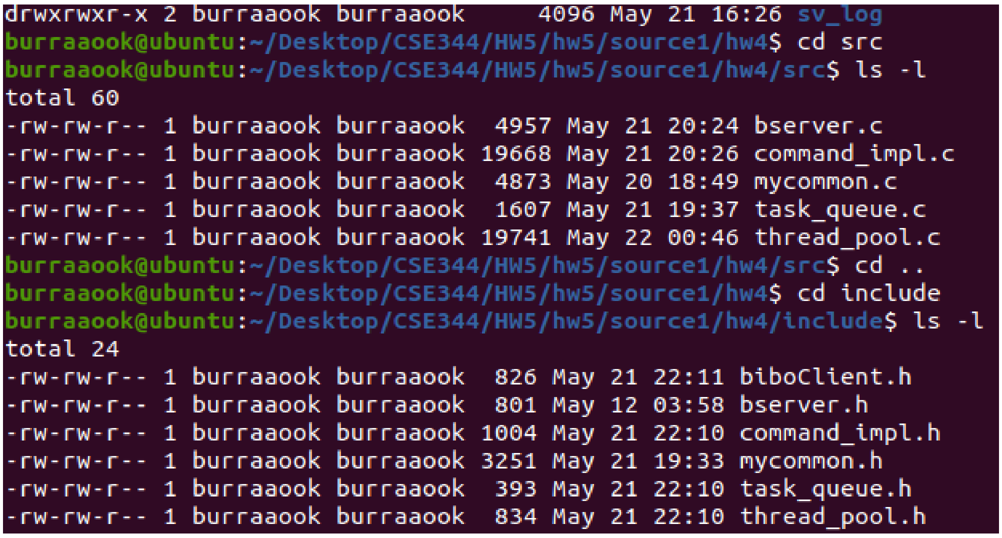
```
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1$ cd HW2
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/HW2$ ls -l
total 640
drwxrwxr-x 3 burraaook burraaook   4096 May 30 00:12 burak_kocausta_1901042605
-rwxrw-rw- 1 burraaook burraaook 647355 May 29 23:02 burak_kocausta_1901042605.zip
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/HW2$ cd burak_kocausta_1901042605
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/HW2/burak_kocausta_1901042605$ ls -l
total 4
drwxrwxr-x 6 burraaook burraaook 4096 May 30 00:12 hw2
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/HW2/burak_kocausta_1901042605$ cd hw2
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/HW2/burak_kocausta_1901042605/hw2$ ls -l
total 56
drwxrwxr-x 2 burraaook burraaook  4096 Apr 14 20:53 include
drwxrwxr-x 2 burraaook burraaook  4096 May 30 00:13 log
-rw-rw-r-- 1 burraaook burraaook   515 Apr 14 20:53 makefile
-rwxrwxr-x 1 burraaook burraaook 28008 May 30 00:12 myshell
-rw-rw-r-- 1 burraaook burraaook  5635 Apr 14 19:46 myshell.c
drwxrwxr-x 2 burraaook burraaook  4096 Apr 14 20:53 report
drwxrwxr-x 2 burraaook burraaook  4096 Apr 14 20:53 src
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/HW2/burak_kocausta_1901042605/hw2$ cd src
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/HW2/burak_kocausta_1901042605/hw2/src$ ls -l
total 24
-rw-rw-r-- 1 burraaook burraaook  1157 Apr 14 04:22 mycommon.c
-rw-rw-r-- 1 burraaook burraaook 16890 Apr 14 19:48 shell_impl.c
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/HW2/burak_kocausta_1901042605/hw2/src$ cd ..
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/HW2/burak_kocausta_1901042605/hw2$ cd include
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/HW2/burak_kocausta_1901042605/hw2/include$ ls -l
total 8
-rw-rw-r-- 1 burraaook burraaook 1119 Apr 14 20:32 mycommon.h
-rw-rw-r-- 1 burraaook burraaook 1455 May 30 08:59 shell_impl.h
```



```
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/HW2/burak_kocausta_1901042605/hw2$ cd log
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/HW2/burak_kocausta_1901042605/hw2/log$ ls -l
total 12
-rw-r--r-- 1 burraaook burraaook 14 May 30 00:12 2023-05-30-00_12_25.txt
-rw-r--r-- 1 burraaook burraaook 20 May 30 00:12 2023-05-30-00_12_27.txt
-rw-r--r-- 1 burraaook burraaook 47 May 30 00:13 2023-05-30-00_13_30.txt
```



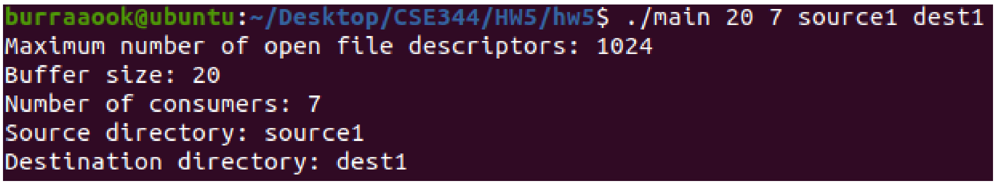
```
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/hw4$ ls -l
total 10804
-rwxrwxr-x 1 burraaook burraaook    31952 May 21 16:21 biboClient
-rw-rw-r-- 1 burraaook burraaook    29879 May 21 20:25 biboClient.c
-rwxrwxr-x 1 burraaook burraaook    51120 May 21 16:21 biboServer
-rw-rw-r-- 1 burraaook burraaook    12630 May 21 22:11 biboServer.c
-rw-rw-r-- 1 burraaook burraaook 10827197 May 21 22:09 book.pdf
-rw-rw-r-- 1 burraaook burraaook      115 May 21 16:51 file1.txt
drwxrwxr-x 2 burraaook burraaook     4096 May 21 16:22 Here
-rw-rw-r-- 1 burraaook burraaook    72869 May 20 17:59 image.png
drwxrwxr-x 2 burraaook burraaook     4096 May 21 20:21 include
-rw-rw-r-- 1 burraaook burraaook      795 May 21 22:15 makefile
drwxrwxr-x 2 burraaook burraaook     4096 May 20 12:55 src
drwxrwxr-x 2 burraaook burraaook     4096 May 21 16:26 sv_log
```



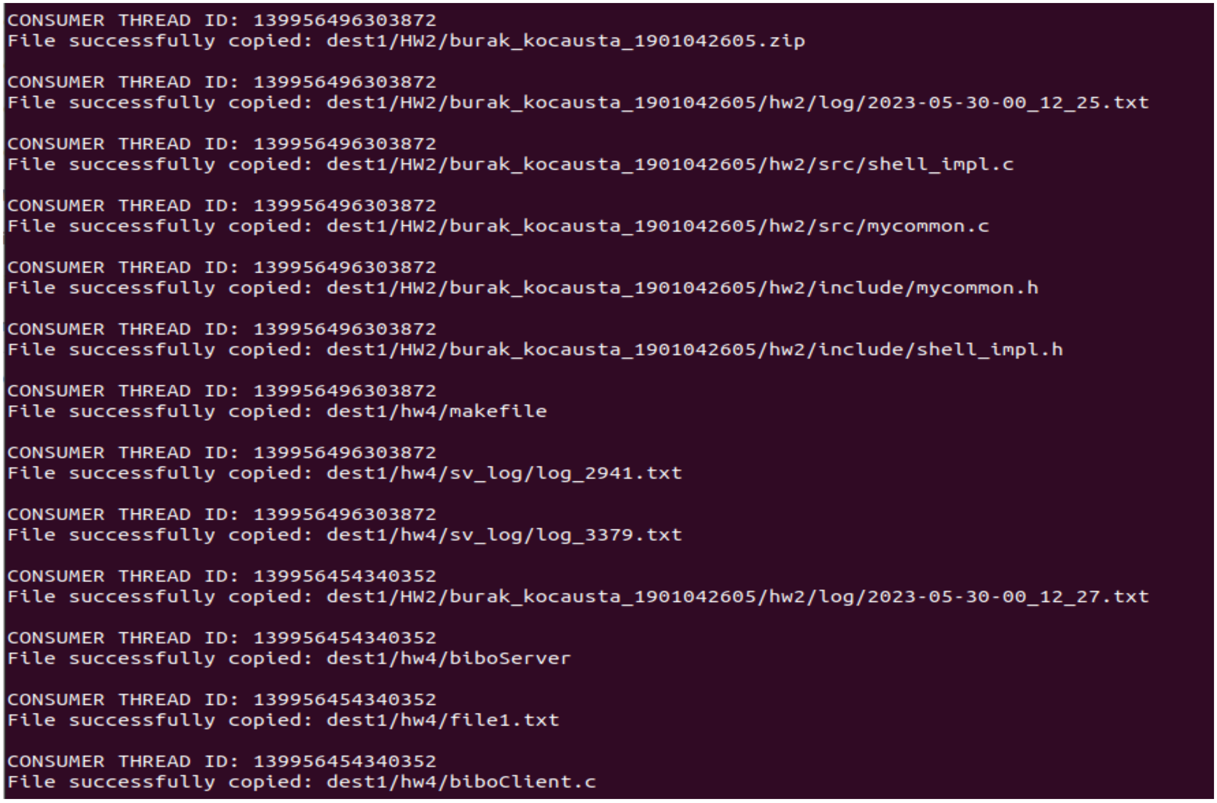
```
drwxrwxr-x 2 burraaook burraaook     4096 May 21 16:26 sv_log
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/hw4$ cd src
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/hw4/src$ ls -l
total 60
-rw-rw-r-- 1 burraaook burraaook  4957 May 21 20:24 bserver.c
-rw-rw-r-- 1 burraaook burraaook 19668 May 21 20:26 command_impl.c
-rw-rw-r-- 1 burraaook burraaook  4873 May 20 18:49 mycommon.c
-rw-rw-r-- 1 burraaook burraaook  1607 May 21 19:37 task_queue.c
-rw-rw-r-- 1 burraaook burraaook 19741 May 22 00:46 thread_pool.c
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/hw4/src$ cd ..
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/hw4$ cd include
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1/hw4/include$ ls -l
total 24
-rw-rw-r-- 1 burraaook burraaook  826 May 21 22:11 biboClient.h
-rw-rw-r-- 1 burraaook burraaook  801 May 12 03:58 bserver.h
-rw-rw-r-- 1 burraaook burraaook 1004 May 21 22:10 command_impl.h
-rw-rw-r-- 1 burraaook burraaook 3251 May 21 19:33 mycommon.h
-rw-rw-r-- 1 burraaook burraaook  393 May 21 22:10 task_queue.h
-rw-rw-r-- 1 burraaook burraaook  834 May 21 22:10 thread_pool.h
```

- Those are the files in this directory.

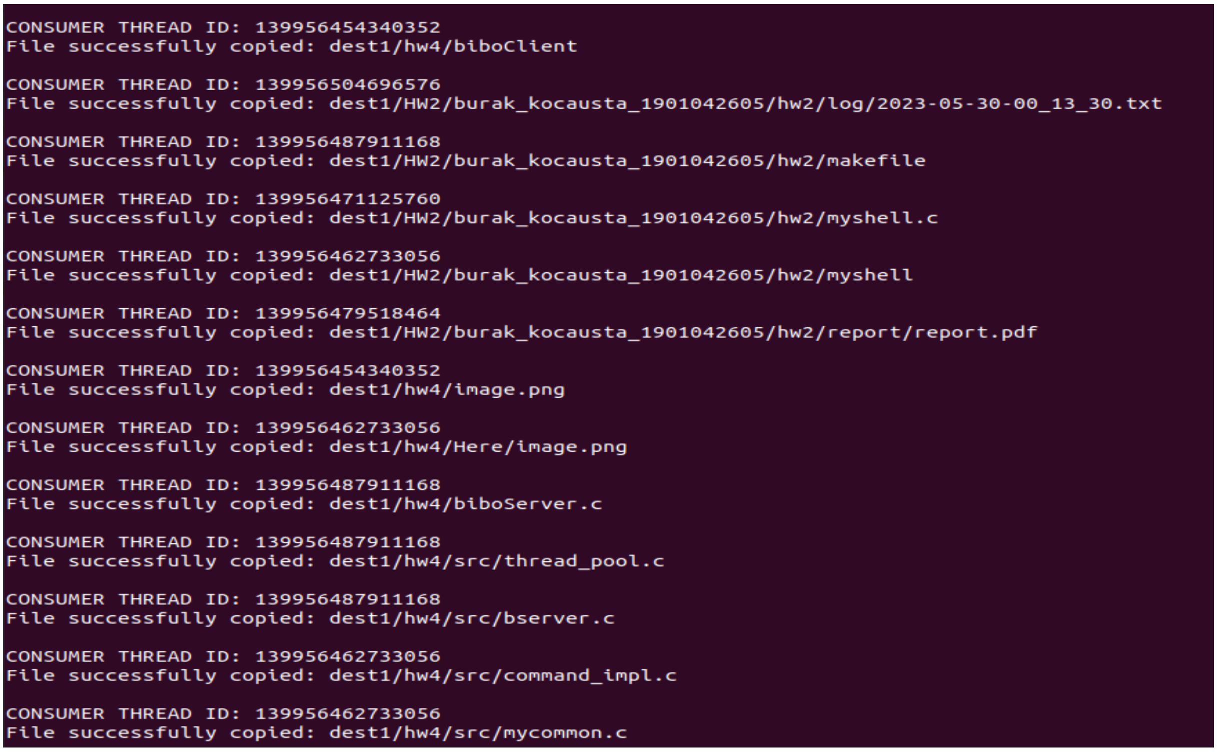
RESULT1:



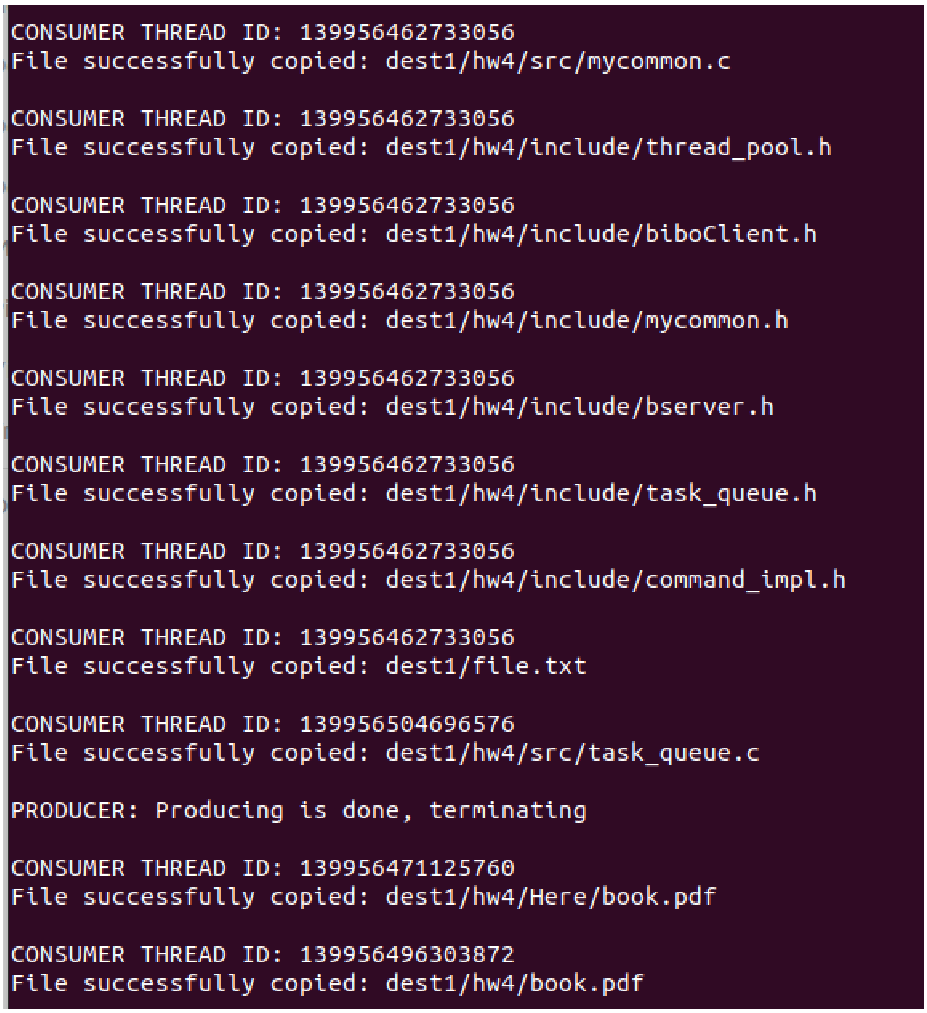
```
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5$ ./main 20 7 source1 dest1
Maximum number of open file descriptors: 1024
Buffer size: 20
Number of consumers: 7
Source directory: source1
Destination directory: dest1
```



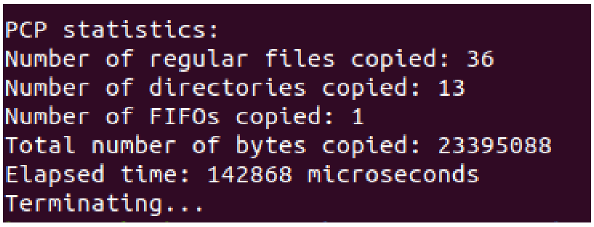
```
CONSUMER THREAD ID: 139956496303872
File successfully copied: dest1/HW2/burak_kocausta_1901042605.zip

CONSUMER THREAD ID: 139956496303872
File successfully copied: dest1/HW2/burak_kocausta_1901042605/hw2/log/2023-05-30-00_12_25.txt

CONSUMER THREAD ID: 139956496303872
File successfully copied: dest1/HW2/burak_kocausta_1901042605/hw2/src/shell_impl.c

CONSUMER THREAD ID: 139956496303872
File successfully copied: dest1/HW2/burak_kocausta_1901042605/hw2/src/mycommon.c

CONSUMER THREAD ID: 139956496303872
File successfully copied: dest1/HW2/burak_kocausta_1901042605/hw2/include/mycommon.h

CONSUMER THREAD ID: 139956496303872
File successfully copied: dest1/HW2/burak_kocausta_1901042605/hw2/include/shell_impl.h

CONSUMER THREAD ID: 139956496303872
File successfully copied: dest1/hw4/makefile

CONSUMER THREAD ID: 139956496303872
File successfully copied: dest1/hw4/sv_log/log_2941.txt

CONSUMER THREAD ID: 139956496303872
File successfully copied: dest1/hw4/sv_log/log_3379.txt

CONSUMER THREAD ID: 139956454340352
File successfully copied: dest1/HW2/burak_kocausta_1901042605/hw2/log/2023-05-30-00_12_27.txt

CONSUMER THREAD ID: 139956454340352
File successfully copied: dest1/hw4/biboServer

CONSUMER THREAD ID: 139956454340352
File successfully copied: dest1/hw4/file1.txt

CONSUMER THREAD ID: 139956454340352
File successfully copied: dest1/hw4/biboClient.c
```



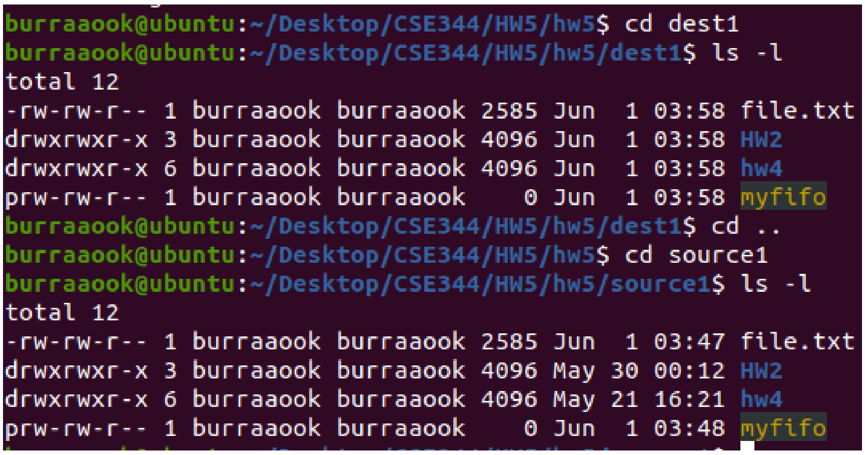
```
CONSUMER THREAD ID: 139956454340352
File successfully copied: dest1/hw4/biboClient

CONSUMER THREAD ID: 139956504696576
File successfully copied: dest1/HW2/burak_kocausta_1901042605/hw2/log/2023-05-30-00_13_30.txt

CONSUMER THREAD ID: 139956487911168
File successfully copied: dest1/HW2/burak_kocausta_1901042605/hw2/makefile

CONSUMER THREAD ID: 139956471125760
File successfully copied: dest1/HW2/burak_kocausta_1901042605/hw2/myshell.c

CONSUMER THREAD ID: 139956462733056
File successfully copied: dest1/HW2/burak_kocausta_1901042605/hw2/myshell

CONSUMER THREAD ID: 139956479518464
File successfully copied: dest1/HW2/burak_kocausta_1901042605/hw2/report/report.pdf

CONSUMER THREAD ID: 139956454340352
File successfully copied: dest1/hw4/image.png

CONSUMER THREAD ID: 139956462733056
File successfully copied: dest1/hw4/Here/image.png

CONSUMER THREAD ID: 139956487911168
File successfully copied: dest1/hw4/biboServer.c

CONSUMER THREAD ID: 139956487911168
File successfully copied: dest1/hw4/src/thread_pool.c

CONSUMER THREAD ID: 139956487911168
File successfully copied: dest1/hw4/src/bserver.c

CONSUMER THREAD ID: 139956462733056
File successfully copied: dest1/hw4/src/command_impl.c

CONSUMER THREAD ID: 139956462733056
File successfully copied: dest1/hw4/src/mycommon.c
```



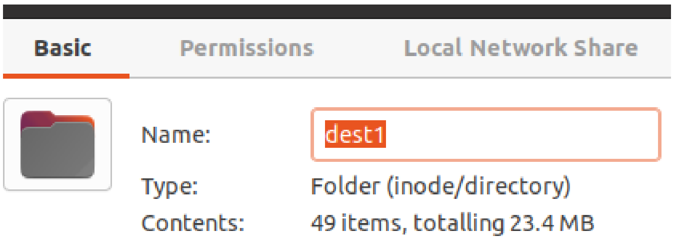
```
CONSUMER THREAD ID: 139956462733056
File successfully copied: dest1/hw4/src/mycommon.c

CONSUMER THREAD ID: 139956462733056
File successfully copied: dest1/hw4/include/thread_pool.h

CONSUMER THREAD ID: 139956462733056
File successfully copied: dest1/hw4/include/biboClient.h

CONSUMER THREAD ID: 139956462733056
File successfully copied: dest1/hw4/include/mycommon.h

CONSUMER THREAD ID: 139956462733056
File successfully copied: dest1/hw4/include/bserver.h

CONSUMER THREAD ID: 139956462733056
File successfully copied: dest1/hw4/include/task_queue.h

CONSUMER THREAD ID: 139956462733056
File successfully copied: dest1/hw4/include/command_impl.h

CONSUMER THREAD ID: 139956462733056
File successfully copied: dest1/file.txt

CONSUMER THREAD ID: 139956504696576
File successfully copied: dest1/hw4/src/task_queue.c

PRODUCER: Producing is done, terminating

CONSUMER THREAD ID: 139956471125760
File successfully copied: dest1/hw4/Here/book.pdf

CONSUMER THREAD ID: 139956496303872
File successfully copied: dest1/hw4/book.pdf
```



```
PCP statistics:
Number of regular files copied: 36
Number of directories copied: 13
Number of FIFOs copied: 1
Total number of bytes copied: 23395088
Elapsed time: 142868 microseconds
Terminating...
```



```
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5$ cd dest1
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/dest1$ ls -l
total 12
-rw-rw-r-- 1 burraaook burraaook 2585 Jun  1 03:58 file.txt
drwxrwxr-x 3 burraaook burraaook 4096 Jun  1 03:58 HW2
drwxrwxr-x 6 burraaook burraaook 4096 Jun  1 03:58 hw4
prw-rw-r-- 1 burraaook burraaook    0 Jun  1 03:58 myfifo
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/dest1$ cd ..
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5$ cd source1
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/source1$ ls -l
total 12
-rw-rw-r-- 1 burraaook burraaook 2585 Jun  1 03:47 file.txt
drwxrwxr-x 3 burraaook burraaook 4096 May 30 00:12 HW2
drwxrwxr-x 6 burraaook burraaook 4096 May 21 16:21 hw4
prw-rw-r-- 1 burraaook burraaook    0 Jun  1 03:48 myfifo
```

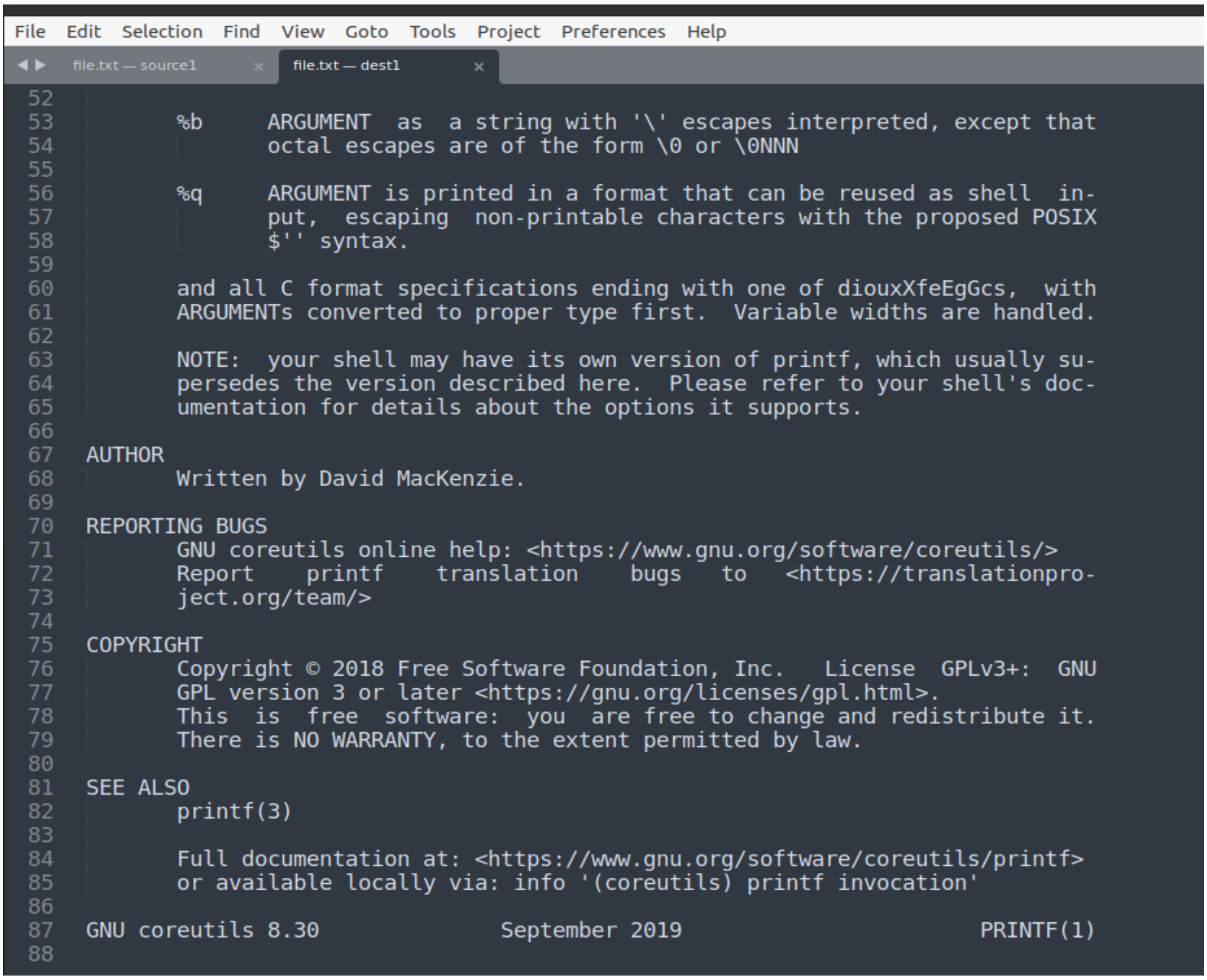


Name: dest1

Type: Folder (inode/directory)

Contents: 49 items, totalling 23.4 MB

- Size is correct.

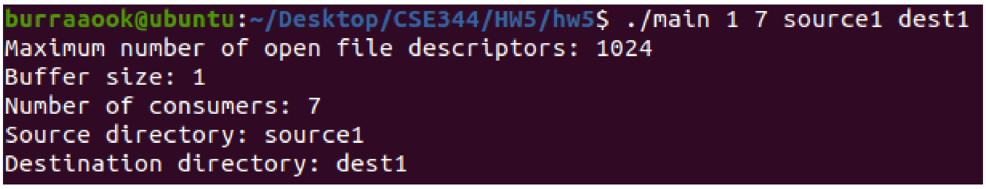


```

       %b     ARGUMENT  as  a string with '\' escapes interpreted, except that
              octal escapes are of the form \0 or \0NNN

       %q     ARGUMENT is printed in a format that can be reused as shell  in-
              put,  escaping  non-printable characters with the proposed POSIX
              $'' syntax.

       and all C format specifications ending with one of diouxXfeEgGcs,  with
       ARGUMENTs converted to proper type first.  Variable widths are handled.

       NOTE:  your shell may have its own version of printf, which usually su-
       persedes the version described here.  Please refer to your shell's doc-
       umentation for details about the options it supports.

AUTHOR
       Written by David MacKenzie.

REPORTING BUGS
       GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
       Report    printf    translation    bugs    to    <https://translationpro-
       ject.org/team/>

COPYRIGHT
       Copyright © 2018 Free Software Foundation, Inc.   License  GPLv3+:  GNU
       GPL version 3 or later <https://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.

SEE ALSO
       printf(3)

       Full documentation at: <https://www.gnu.org/software/coreutils/printf>
       or available locally via: info '(coreutils) printf invocation'

GNU coreutils 8.30              September 2019                       PRINTF(1)

```

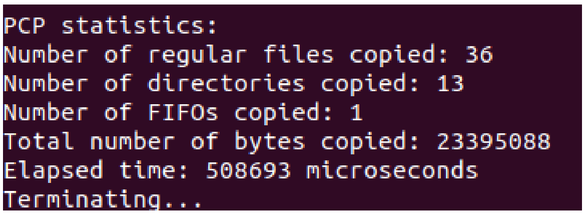
- The file is same, no data corruption.

TEST 2:

- Testing same size with different small buffer size.



```
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5$ ./main 1 7 source1 dest1
Maximum number of open file descriptors: 1024
Buffer size: 1
Number of consumers: 7
Source directory: source1
Destination directory: dest1
```

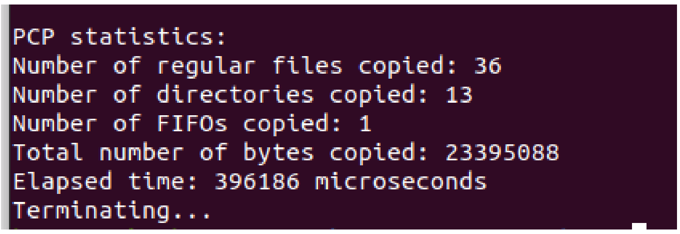


```
PCP statistics:
Number of regular files copied: 36
Number of directories copied: 13
Number of FIFOs copied: 1
Total number of bytes copied: 23395088
Elapsed time: 508693 microseconds
Terminating...
```

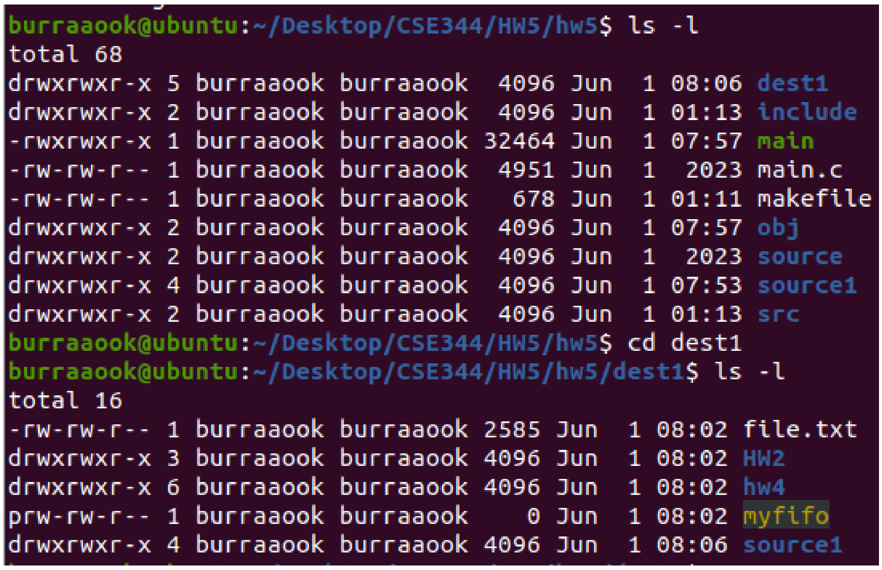
- It takes much longer.

TEST 3:

- Use same source and destination again. It must create source1 directory inside dest1 directory. Identical to the cp -R command.



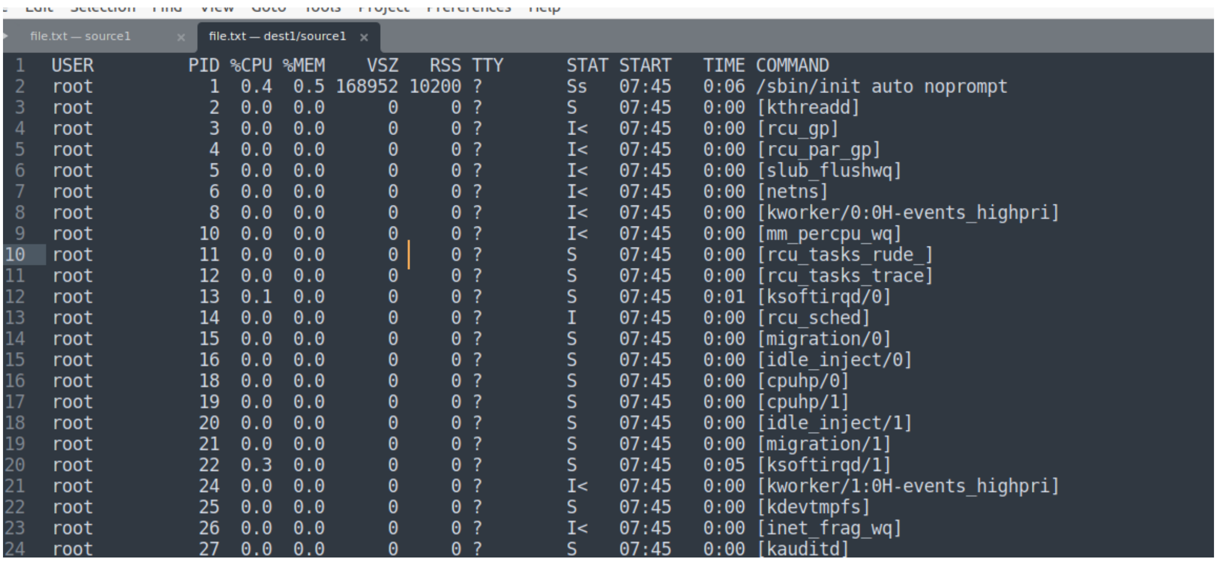
```
PCP statistics:
Number of regular files copied: 36
Number of directories copied: 13
Number of FIFOs copied: 1
Total number of bytes copied: 23395088
Elapsed time: 396186 microseconds
Terminating...
```



```
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5$ ls -l
total 68
drwxrwxr-x 5 burraaook burraaook  4096 Jun  1 08:06 dest1
drwxrwxr-x 2 burraaook burraaook  4096 Jun  1 01:13 include
-rwxrwxr-x 1 burraaook burraaook 32464 Jun  1 07:57 main
-rw-rw-r-- 1 burraaook burraaook  4951 Jun  1  2023 main.c
-rw-rw-r-- 1 burraaook burraaook   678 Jun  1 01:11 makefile
drwxrwxr-x 2 burraaook burraaook  4096 Jun  1 07:57 obj
drwxrwxr-x 2 burraaook burraaook  4096 Jun  1  2023 source
drwxrwxr-x 4 burraaook burraaook  4096 Jun  1 07:53 source1
drwxrwxr-x 2 burraaook burraaook  4096 Jun  1 01:13 src
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5$ cd dest1
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5/dest1$ ls -l
total 16
-rw-rw-r-- 1 burraaook burraaook 2585 Jun  1 08:02 file.txt
drwxrwxr-x 3 burraaook burraaook 4096 Jun  1 08:02 HW2
drwxrwxr-x 6 burraaook burraaook 4096 Jun  1 08:02 hw4
prw-rw-r-- 1 burraaook burraaook    0 Jun  1 08:02 myfifo
drwxrwxr-x 4 burraaook burraaook 4096 Jun  1 08:06 source1
```

TEST 4:

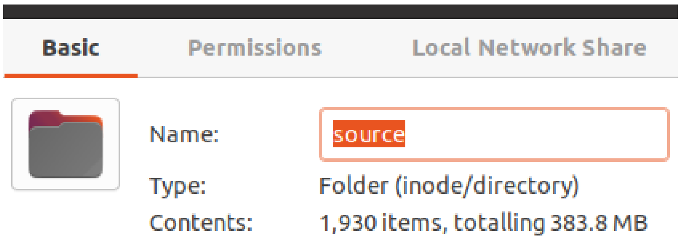
- Updating file.txt in source1 file. test if it updates in the destination.



```
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.4  0.5 168952 10200 ?        Ss   07:45   0:06 /sbin/init auto noprompt
root           2  0.0  0.0      0     0 ?        S    07:45   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   07:45   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   07:45   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   07:45   0:00 [slub_flushwq]
root           6  0.0  0.0      0     0 ?        I<   07:45   0:00 [netns]
root           8  0.0  0.0      0     0 ?        I<   07:45   0:00 [kworker/0:0H-events_highpri]
root          10  0.0  0.0      0     0 ?        I<   07:45   0:00 [mm_percpu_wq]
root          11  0.0  0.0      0     0 ?        S    07:45   0:00 [rcu_tasks_rude_]
root          12  0.0  0.0      0     0 ?        S    07:45   0:00 [rcu_tasks_trace]
root          13  0.1  0.0      0     0 ?        S    07:45   0:01 [ksoftirqd/0]
root          14  0.0  0.0      0     0 ?        I    07:45   0:00 [rcu_sched]
root          15  0.0  0.0      0     0 ?        S    07:45   0:00 [migration/0]
root          16  0.0  0.0      0     0 ?        S    07:45   0:00 [idle_inject/0]
root          18  0.0  0.0      0     0 ?        S    07:45   0:00 [cpuhp/0]
root          19  0.0  0.0      0     0 ?        S    07:45   0:00 [cpuhp/1]
root          20  0.0  0.0      0     0 ?        S    07:45   0:00 [idle_inject/1]
root          21  0.0  0.0      0     0 ?        S    07:45   0:00 [migration/1]
root          22  0.3  0.0      0     0 ?        S    07:45   0:05 [ksoftirqd/1]
root          24  0.0  0.0      0     0 ?        I<   07:45   0:00 [kworker/1:0H-events_highpri]
root          25  0.0  0.0      0     0 ?        S    07:45   0:00 [kdevtmpfs]
root          26  0.0  0.0      0     0 ?        I<   07:45   0:00 [inet_frag_wq]
root          27  0.0  0.0      0     0 ?        S    07:45   0:00 [kauditd]
```

- It is updated.

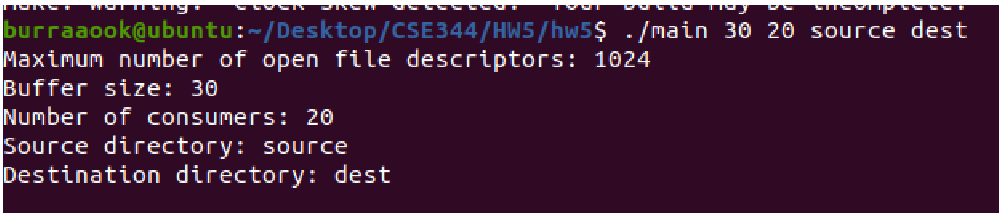
- Try larger files



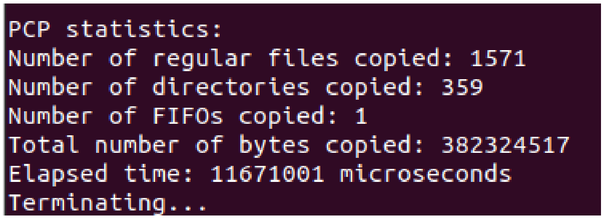
Name: source

Type: Folder (inode/directory)

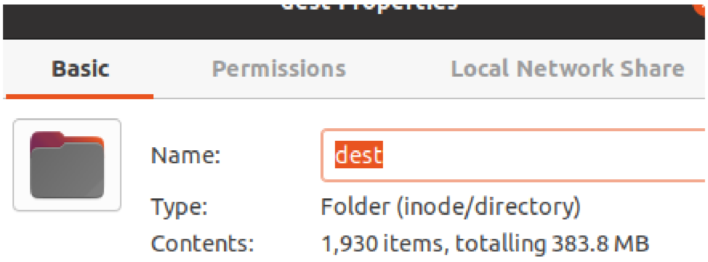
Contents: 1,930 items, totalling 383.8 MB



```
make: warning:  Clock skew detected.  Your build may be incomplete.
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5$ ./main 30 20 source dest
Maximum number of open file descriptors: 1024
Buffer size: 30
Number of consumers: 20
Source directory: source
Destination directory: dest
```



```
PCP statistics:
Number of regular files copied: 1571
Number of directories copied: 359
Number of FIFOs copied: 1
Total number of bytes copied: 382324517
Elapsed time: 11671001 microseconds
Terminating...
```

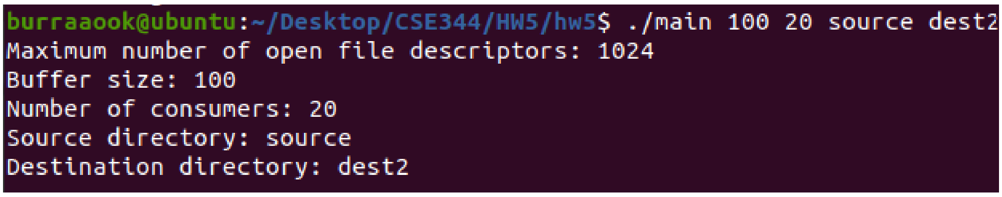


Name: dest

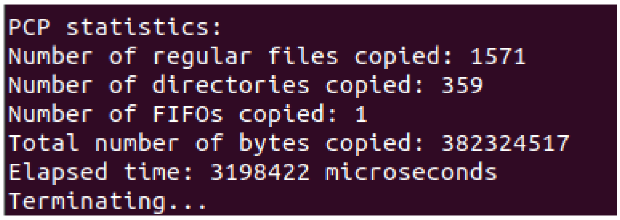
Type: Folder (inode/directory)

Contents: 1,930 items, totalling 383.8 MB

Try with different buffer size:



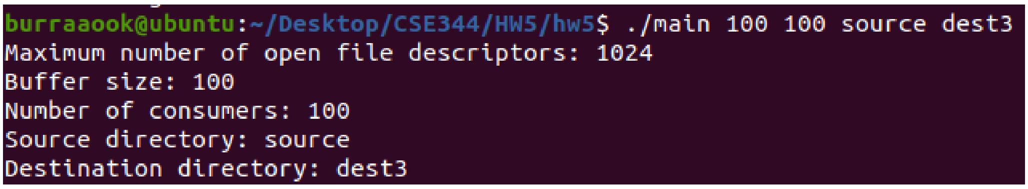
```
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5$ ./main 100 20 source dest2
Maximum number of open file descriptors: 1024
Buffer size: 100
Number of consumers: 20
Source directory: source
Destination directory: dest2
```



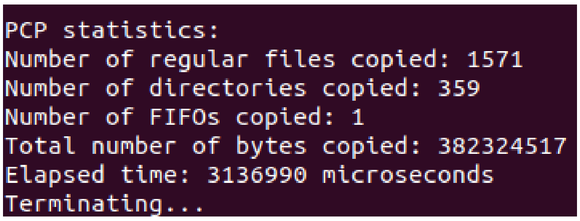
```
PCP statistics:
Number of regular files copied: 1571
Number of directories copied: 359
Number of FIFOs copied: 1
Total number of bytes copied: 382324517
Elapsed time: 3198422 microseconds
Terminating...
```

- With larger buffer size, it significantly takes less time.

Try with different thread size:



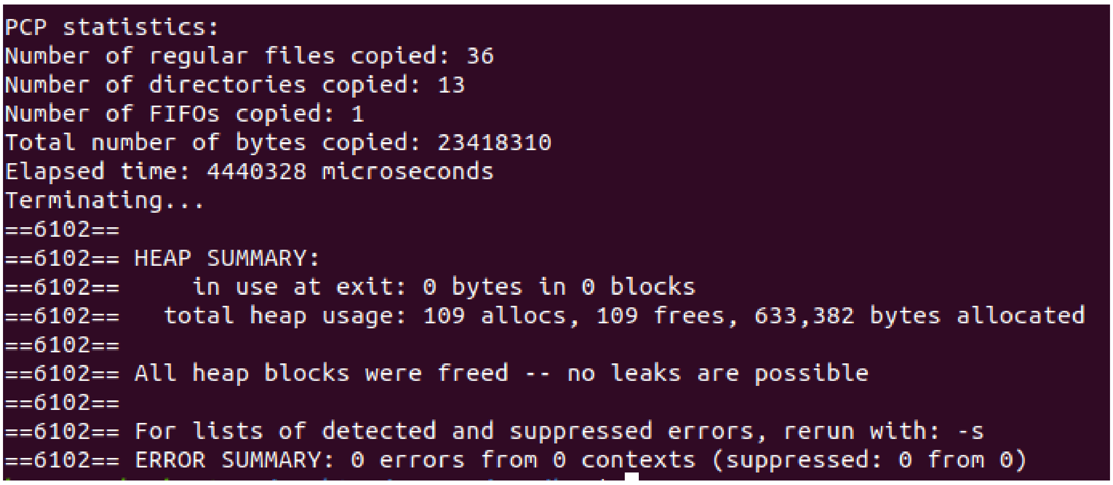
```
burraaook@ubuntu:~/Desktop/CSE344/HW5/hw5$ ./main 100 100 source dest3
Maximum number of open file descriptors: 1024
Buffer size: 100
Number of consumers: 100
Source directory: source
Destination directory: dest3
```



```
PCP statistics:
Number of regular files copied: 1571
Number of directories copied: 359
Number of FIFOs copied: 1
Total number of bytes copied: 382324517
Elapsed time: 3136990 microseconds
Terminating...
```

- It takes less time if consumer thread size increases.

MEMORY CHECK:



```
PCP statistics:
Number of regular files copied: 36
Number of directories copied: 13
Number of FIFOs copied: 1
Total number of bytes copied: 23418310
Elapsed time: 4440328 microseconds
Terminating...
==6102==
==6102== HEAP SUMMARY:
==6102==     in use at exit: 0 bytes in 0 blocks
==6102==   total heap usage: 109 allocs, 109 frees, 633,382 bytes allocated
==6102==
==6102== All heap blocks were freed -- no leaks are possible
==6102==
==6102== For lists of detected and suppressed errors, rerun with: -s
==6102== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

- Memory results of copying source1 directory.