SYSTEM CALL PROGRAMS Muhammed Nifal V

Assignment 5: CB.EN.P2CYS23017

1. Write your own version of printf named myprintfunction().

a. It should be able to accept various types of parameters such as char, int, double, etc.

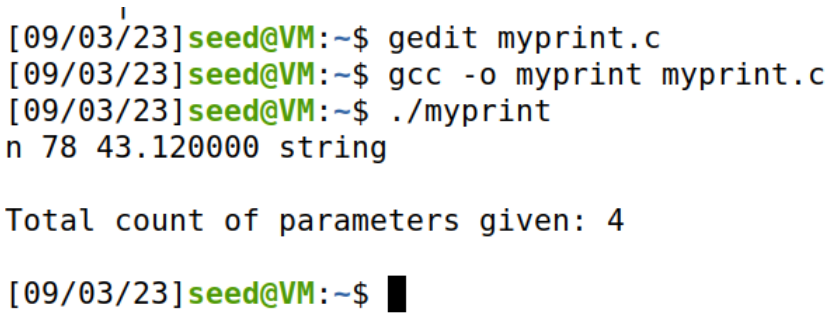
b. Bonus : The function should be able to accept different parameter count. The first parameter says the count of parameters, followed by actual parameters

```
#include <stdio.h>
#include <stdarg.h>
void myprintf(const char *format, ...)
{
   va_list args;
   va_start(args, format);
   int paramCount = 0;
   while (*format)
   {
      if (*format == '%')
      {
         format++;
         switch (*format)
         {
            case 'c':
               paramCount++;
               putchar(va_arg(args, int));
```

```
                break;
            case 'd':
                paramCount++;
                printf("%d", va_arg(args, int));
                break;
            case 'f':
                paramCount++;
                printf("%f", va_arg(args, double));
                break;
            case 's':
                paramCount++;
                fputs(va_arg(args, const char*), stdout);
                break;
            default:
                putchar(*format);
                break;
          }
       }
       else
          putchar(*format);
       format++;
    }
    va_end(args);
    printf("\nTotal count of parameters given: %d\n\n", paramCount);
}
int main()
```

```
{
   myprintf("%c   %d %f %s\n", 'z', 54, 87.28, "New String");
   return 0;
}
```

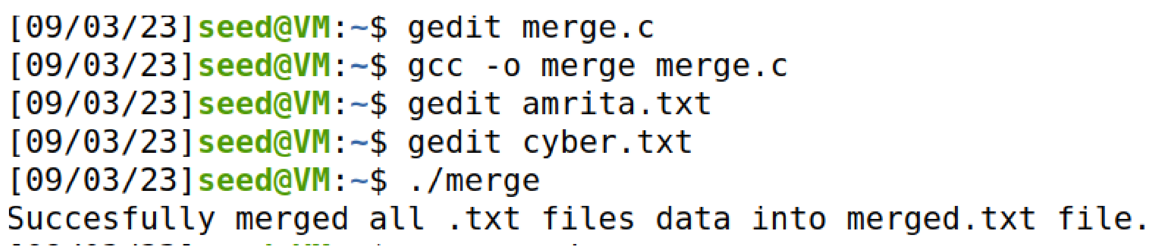
OUTPUT:



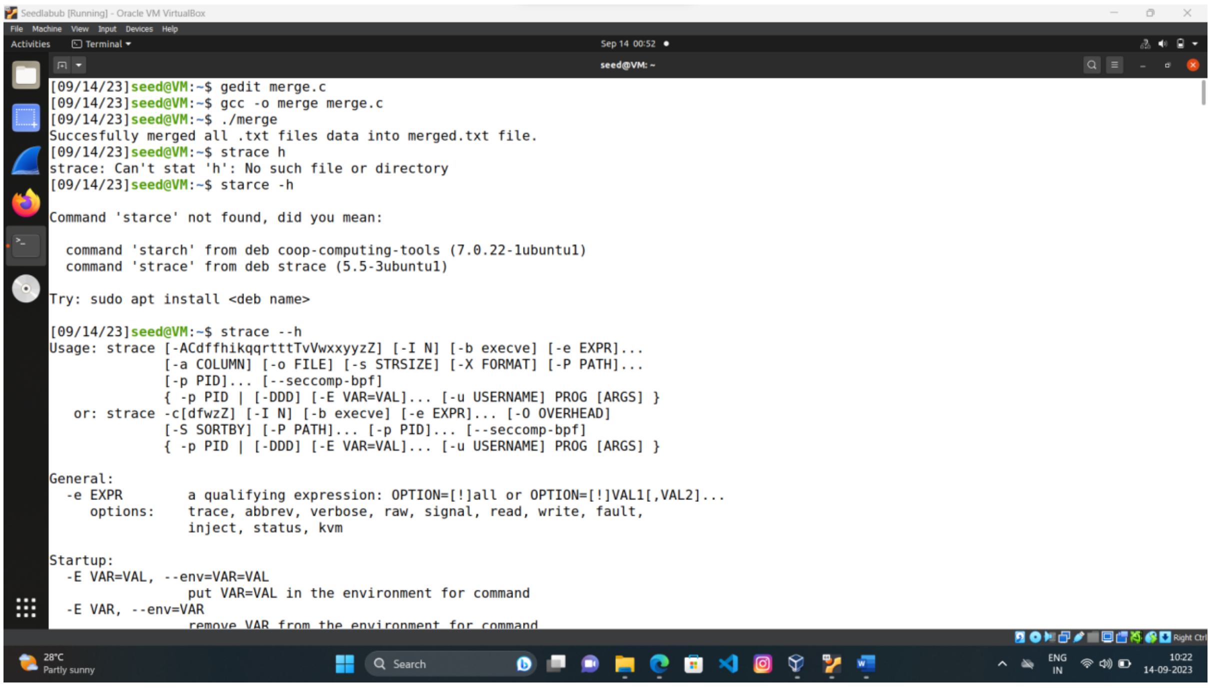
```
[09/03/23]seed@VM:~$ gedit myprint.c
[09/03/23]seed@VM:~$ gcc -o myprint myprint.c
[09/03/23]seed@VM:~$ ./myprint
n 78 43.120000 string

Total count of parameters given: 4

[09/03/23]seed@VM:~$ 
```

2.Write a program to read all txt files (that is files that ends with .txt) in the current directory and merge them all to one text file and return a file descriptor for the newfile.

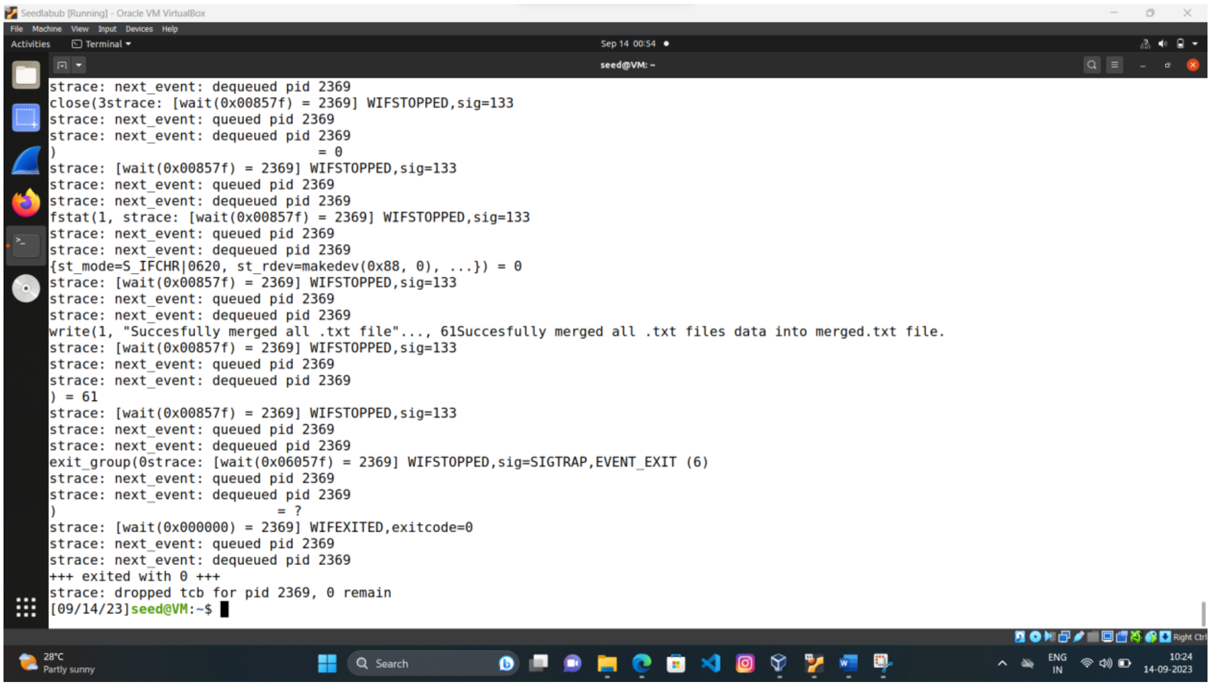
```
#include <stdio.h>
#include <dirent.h>
#include <string.h>
int main(void)
{
FILE *ip, *op;
char ch;
char *txt = ".txt";
struct dirent *de;
DIR *dir = opendir(".");
if(dir == NULL)
{
printf("Can't open current directory.");
return 0;
}
while((de = readdir(dir)) != NULL)
{
char *filename = de->d_name;
char *ext = strrchr(filename, '.');
if(!(!ext || ext == filename))
{
if(strcmp(ext, txt) == 0)
{
op = fopen("merged.txt", "a+");
ip = fopen(filename, "r");
while(1)
{
ch = fgetc(ip);
if(ch == EOF)
break;
putc(ch, op);
}
fclose(ip);
fclose(op);
}
}
}
closedir(dir);
printf("Succesfully merged all .txt files data into merged.txt file.\n");
return 0;
}
```



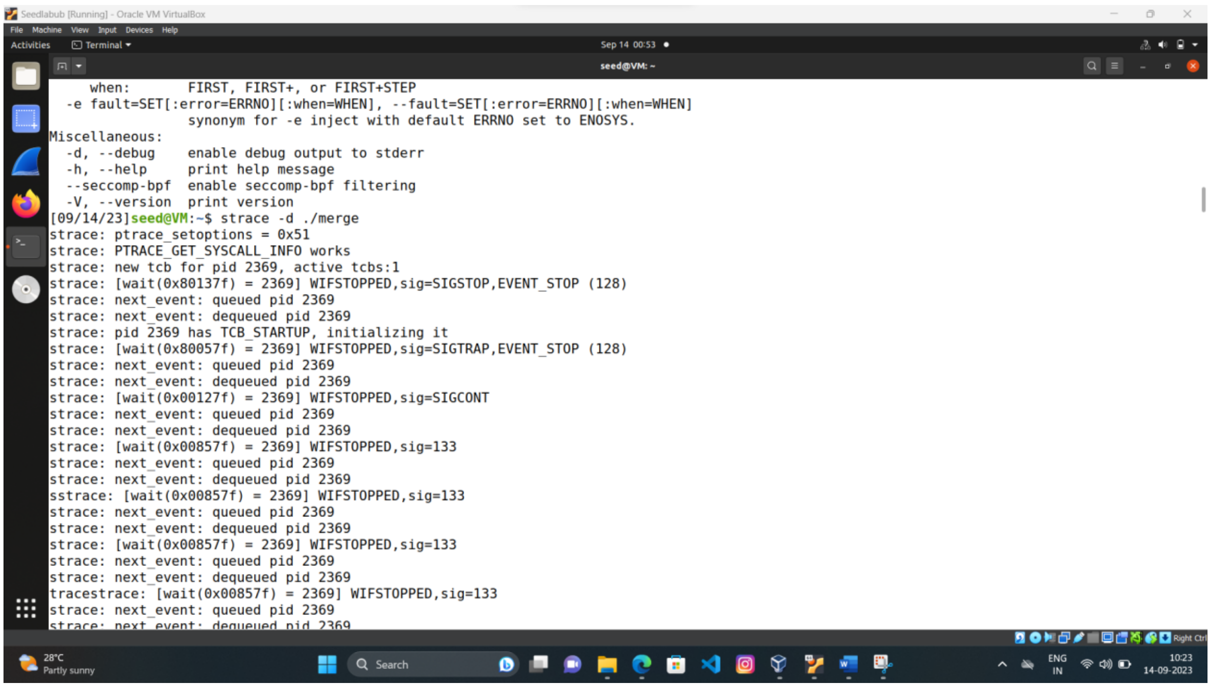
```
[09/03/23]seed@VM:~$ gedit merge.c
[09/03/23]seed@VM:~$ gcc -o merge merge.c
[09/03/23]seed@VM:~$ gedit amrita.txt
[09/03/23]seed@VM:~$ gedit cyber.txt
[09/03/23]seed@VM:~$ ./merge
Succesfully merged all .txt files data into merged.txt file.
```



```
[09/14/23]seed@VM:~$ gedit merge.c
[09/14/23]seed@VM:~$ gcc -o merge merge.c
[09/14/23]seed@VM:~$ ./merge
Succesfully merged all .txt files data into merged.txt file.
[09/14/23]seed@VM:~$ strace h
strace: Can't stat 'h': No such file or directory
[09/14/23]seed@VM:~$ starce -h

Command 'starce' not found, did you mean:

  command 'starch' from deb coop-computing-tools (7.0.22-1ubuntu1)
  command 'strace' from deb strace (5.5-3ubuntu1)

Try: sudo apt install <deb name>

[09/14/23]seed@VM:~$ strace --h
Usage: strace [-ACdffhikqqrtttTvVwxxyyzZ] [-I N] [-b execve] [-e EXPR]...
              [-a COLUMN] [-o FILE] [-s STRSIZE] [-X FORMAT] [-P PATH]...
              [-p PID]... [--seccomp-bpf]
              { -p PID | [-DDD] [-E VAR=VAL]... [-u USERNAME] PROG [ARGS] }
   or: strace -c[dfwzZ] [-I N] [-b execve] [-e EXPR]... [-O OVERHEAD]
              [-S SORTBY] [-P PATH]... [-p PID]... [--seccomp-bpf]
              { -p PID | [-DDD] [-E VAR=VAL]... [-u USERNAME] PROG [ARGS] }

General:
  -e EXPR        a qualifying expression: OPTION=[!]all or OPTION=[!]VAL1[,VAL2]...
     options:    trace, abbrev, verbose, raw, signal, read, write, fault,
                 inject, status, kvm

Startup:
  -E VAR=VAL, --env=VAR=VAL
                 put VAR=VAL in the environment for command
  -E VAR, --env=VAR
                 remove VAR from the environment for command
```



```
strace: next_event: dequeued pid 2369
close(3strace: [wait(0x00857f) = 2369] WIFSTOPPED,sig=133
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
)                                       = 0
strace: [wait(0x00857f) = 2369] WIFSTOPPED,sig=133
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
fstat(1, strace: [wait(0x00857f) = 2369] WIFSTOPPED,sig=133
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
{st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
strace: [wait(0x00857f) = 2369] WIFSTOPPED,sig=133
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
write(1, "Succesfully merged all .txt file"..., 61Succesfully merged all .txt files data into merged.txt file.
strace: [wait(0x00857f) = 2369] WIFSTOPPED,sig=133
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
) = 61
strace: [wait(0x00857f) = 2369] WIFSTOPPED,sig=133
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
exit_group(0strace: [wait(0x06057f) = 2369] WIFSTOPPED,sig=SIGTRAP,EVENT_EXIT (6)
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
)                                       = ?
strace: [wait(0x000000) = 2369] WIFEXITED,exitcode=0
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
+++ exited with 0 +++
strace: dropped tcb for pid 2369, 0 remain
[09/14/23]seed@VM:~$
```



```
    when:        FIRST, FIRST+, or FIRST+STEP
  -e fault=SET[:error=ERRNO][:when=WHEN], --fault=SET[:error=ERRNO][:when=WHEN]
                 synonym for -e inject with default ERRNO set to ENOSYS.
Miscellaneous:
  -d, --debug    enable debug output to stderr
  -h, --help     print help message
  --seccomp-bpf  enable seccomp-bpf filtering
  -V, --version  print version
[09/14/23]seed@VM:~$ strace -d ./merge
strace: ptrace_setoptions = 0x51
strace: PTRACE_GET_SYSCALL_INFO works
strace: new tcb for pid 2369, active tcbs:1
strace: [wait(0x80137f) = 2369] WIFSTOPPED,sig=SIGSTOP,EVENT_STOP (128)
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
strace: pid 2369 has TCB_STARTUP, initializing it
strace: [wait(0x80057f) = 2369] WIFSTOPPED,sig=SIGTRAP,EVENT_STOP (128)
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
strace: [wait(0x00127f) = 2369] WIFSTOPPED,sig=SIGCONT
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
strace: [wait(0x00857f) = 2369] WIFSTOPPED,sig=133
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
sstrace: [wait(0x00857f) = 2369] WIFSTOPPED,sig=133
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
strace: [wait(0x00857f) = 2369] WIFSTOPPED,sig=133
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
tracestrace: [wait(0x00857f) = 2369] WIFSTOPPED,sig=133
strace: next_event: queued pid 2369
strace: next_event: dequeued pid 2369
```

<u>Strace:</u>

<u>Strace is mainly used for following functions</u>

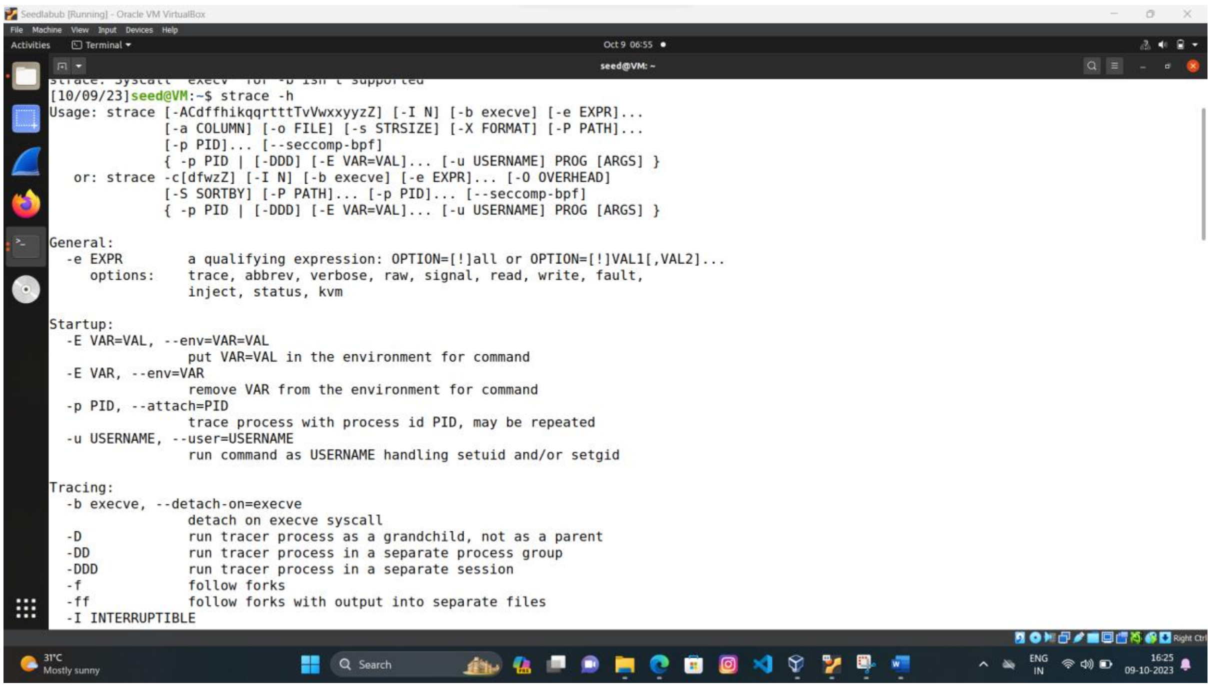
1. Debugging Programs: helps for troubleshooting issues by showing how a program interacts with the system.
2. Troubleshooting Programs: memory leaks
3. Intercepting system calls by a process: It traces all system calls issued by a program along with their return codes.
4. Recording system calls by a process: It returns the name of each system call along with its argument.
5. Process Monitoring: It allows to find out how a program is interacting with the OS.

Eg: strace -d : print debugging output

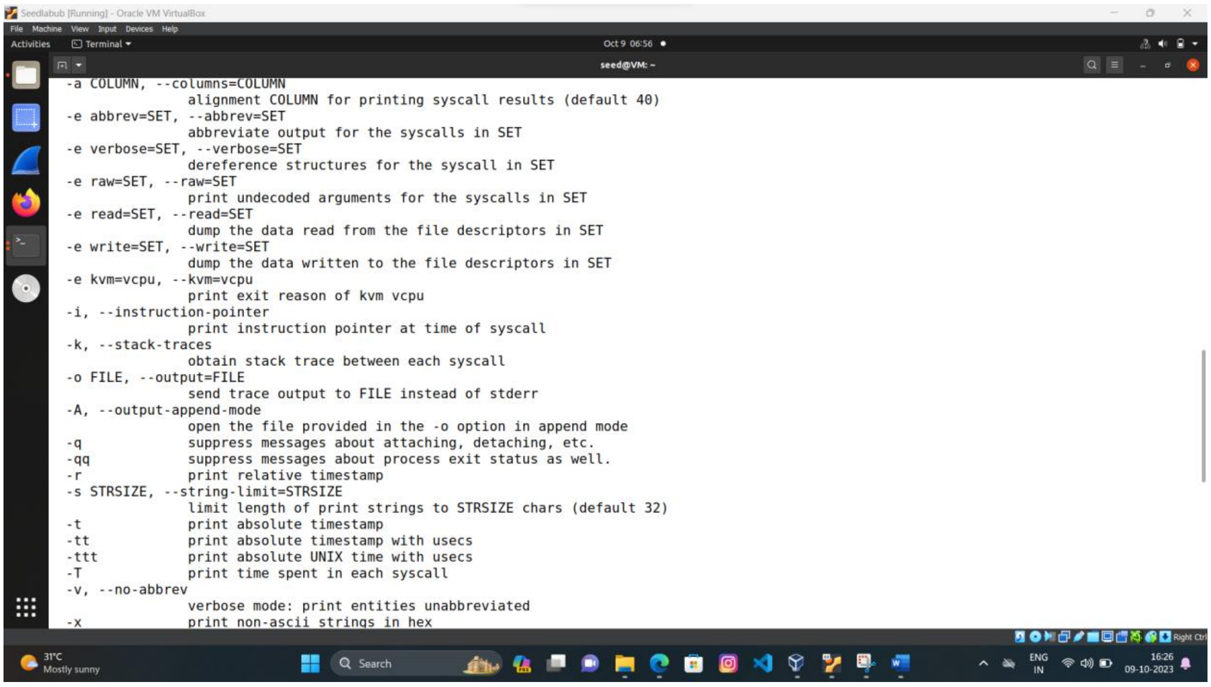
Strace -h : for help

Starce -c: for displaying system calls, no: of times system calls called.

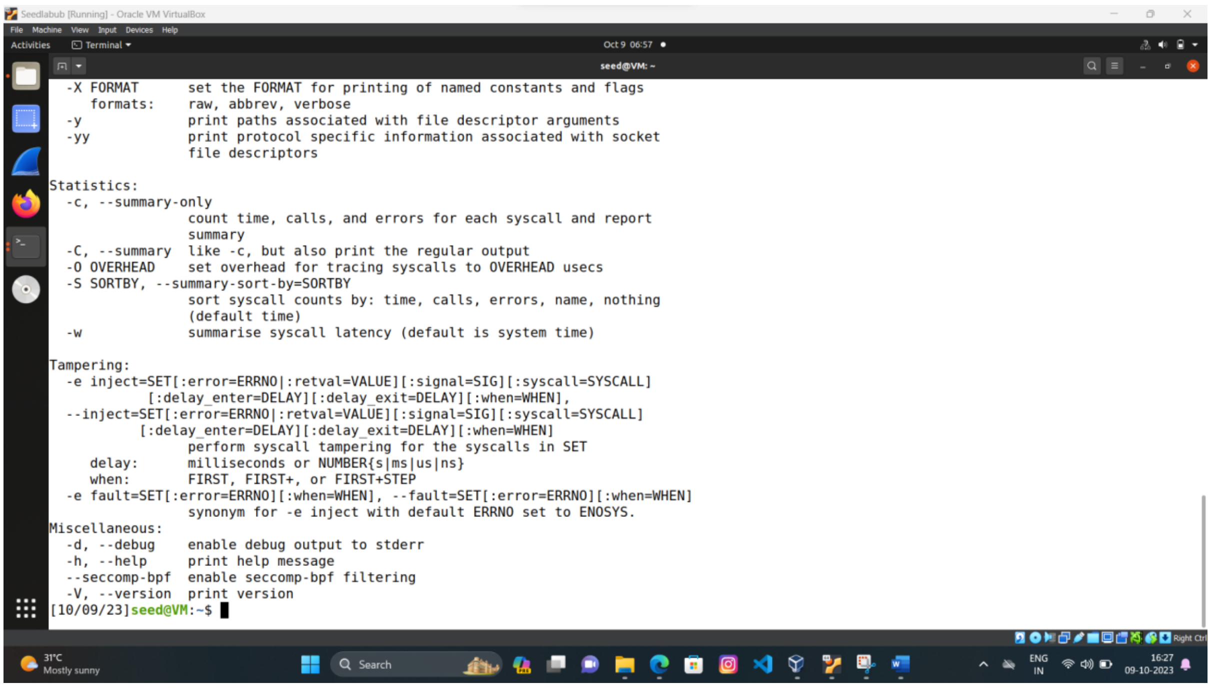
Strace -h



```
strace: syscall execv for -b isn't supported
[10/09/23]seed@VM:~$ strace -h
Usage: strace [-ACdffhikqqrtttTvVwxxyyzZ] [-I N] [-b execve] [-e EXPR]...
              [-a COLUMN] [-o FILE] [-s STRSIZE] [-X FORMAT] [-P PATH]...
              [-p PID]... [--seccomp-bpf]
              { -p PID | [-DDD] [-E VAR=VAL]... [-u USERNAME] PROG [ARGS] }
   or: strace -c[dfwzZ] [-I N] [-b execve] [-e EXPR]... [-O OVERHEAD]
              [-S SORTBY] [-P PATH]... [-p PID]... [--seccomp-bpf]
              { -p PID | [-DDD] [-E VAR=VAL]... [-u USERNAME] PROG [ARGS] }

General:
  -e EXPR        a qualifying expression: OPTION=[!]all or OPTION=[!]VAL1[,VAL2]...
     options:    trace, abbrev, verbose, raw, signal, read, write, fault,
                 inject, status, kvm

Startup:
  -E VAR=VAL, --env=VAR=VAL
                 put VAR=VAL in the environment for command
  -E VAR, --env=VAR
                 remove VAR from the environment for command
  -p PID, --attach=PID
                 trace process with process id PID, may be repeated
  -u USERNAME, --user=USERNAME
                 run command as USERNAME handling setuid and/or setgid

Tracing:
  -b execve, --detach-on=execve
                 detach on execve syscall
  -D             run tracer process as a grandchild, not as a parent
  -DD            run tracer process in a separate process group
  -DDD           run tracer process in a separate session
  -f             follow forks
  -ff            follow forks with output into separate files
  -I INTERRUPTIBLE
```

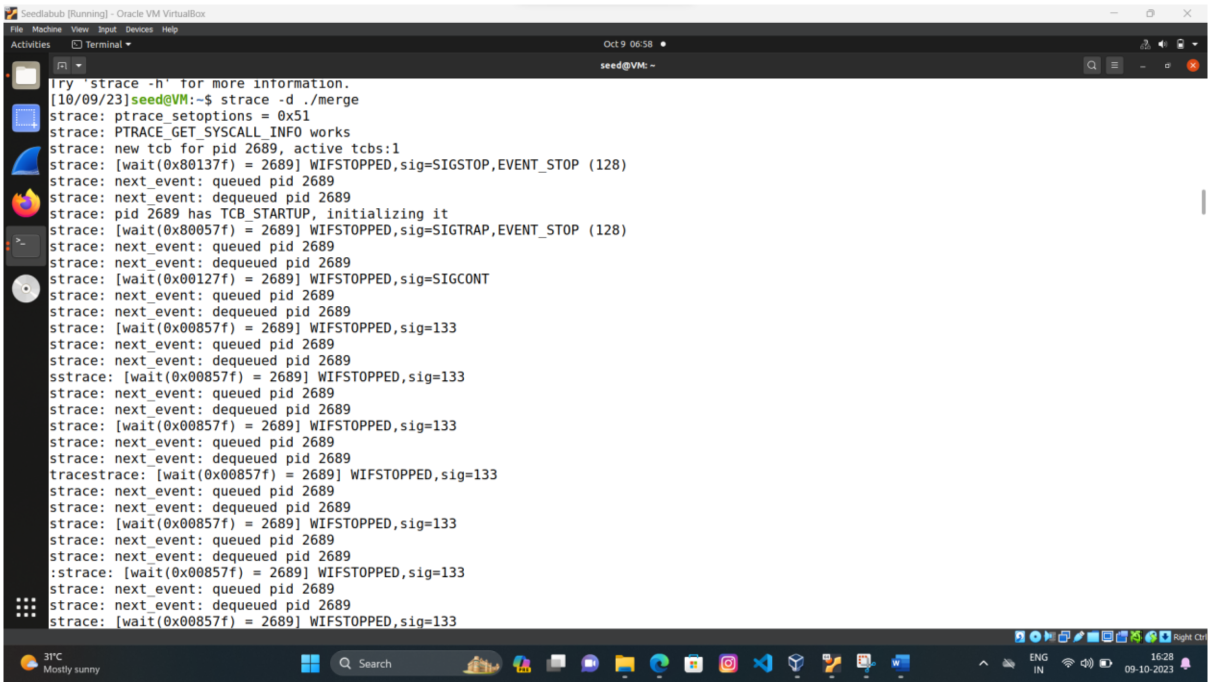


```
  -a COLUMN, --columns=COLUMN
                 alignment COLUMN for printing syscall results (default 40)
  -e abbrev=SET, --abbrev=SET
                 abbreviate output for the syscalls in SET
  -e verbose=SET, --verbose=SET
                 dereference structures for the syscall in SET
  -e raw=SET, --raw=SET
                 print undecoded arguments for the syscalls in SET
  -e read=SET, --read=SET
                 dump the data read from the file descriptors in SET
  -e write=SET, --write=SET
                 dump the data written to the file descriptors in SET
  -e kvm=vcpu, --kvm=vcpu
                 print exit reason of kvm vcpu
  -i, --instruction-pointer
                 print instruction pointer at time of syscall
  -k, --stack-traces
                 obtain stack trace between each syscall
  -o FILE, --output=FILE
                 send trace output to FILE instead of stderr
  -A, --output-append-mode
                 open the file provided in the -o option in append mode
  -q             suppress messages about attaching, detaching, etc.
  -qq            suppress messages about process exit status as well.
  -r             print relative timestamp
  -s STRSIZE, --string-limit=STRSIZE
                 limit length of print strings to STRSIZE chars (default 32)
  -t             print absolute timestamp
  -tt            print absolute timestamp with usecs
  -ttt           print absolute UNIX time with usecs
  -T             print time spent in each syscall
  -v, --no-abbrev
                 verbose mode: print entities unabbreviated
  -x             print non-ascii strings in hex
```



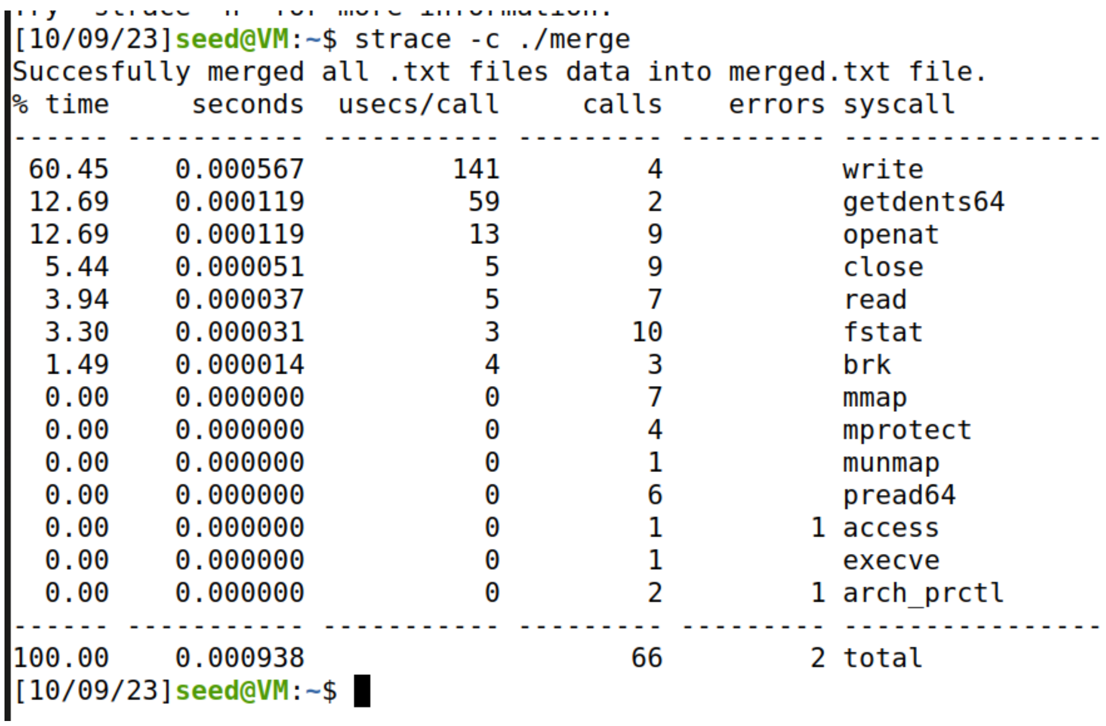
```
  -X FORMAT      set the FORMAT for printing of named constants and flags
     formats:    raw, abbrev, verbose
  -y             print paths associated with file descriptor arguments
  -yy            print protocol specific information associated with socket
                 file descriptors

Statistics:
  -c, --summary-only
                 count time, calls, and errors for each syscall and report
                 summary
  -C, --summary  like -c, but also print the regular output
  -O OVERHEAD    set overhead for tracing syscalls to OVERHEAD usecs
  -S SORTBY, --summary-sort-by=SORTBY
                 sort syscall counts by: time, calls, errors, name, nothing
                 (default time)
  -w             summarise syscall latency (default is system time)

Tampering:
  -e inject=SET[:error=ERRNO|:retval=VALUE][:signal=SIG][:syscall=SYSCALL]
            [:delay_enter=DELAY][:delay_exit=DELAY][:when=WHEN],
  --inject=SET[:error=ERRNO|:retval=VALUE][:signal=SIG][:syscall=SYSCALL]
           [:delay_enter=DELAY][:delay_exit=DELAY][:when=WHEN]
                 perform syscall tampering for the syscalls in SET
     delay:      milliseconds or NUMBER{s|ms|us|ns}
     when:       FIRST, FIRST+, or FIRST+STEP
  -e fault=SET[:error=ERRNO][:when=WHEN], --fault=SET[:error=ERRNO][:when=WHEN]
                 synonym for -e inject with default ERRNO set to ENOSYS.
Miscellaneous:
  -d, --debug    enable debug output to stderr
  -h, --help     print help message
  --seccomp-bpf  enable seccomp-bpf filtering
  -V, --version  print version
[10/09/23]seed@VM:~$
```

Strace -d ./merge



```
Try 'strace -h' for more information.
[10/09/23]seed@VM:~$ strace -d ./merge
strace: ptrace_setoptions = 0x51
strace: PTRACE_GET_SYSCALL_INFO works
strace: new tcb for pid 2689, active tcbs:1
strace: [wait(0x80137f) = 2689] WIFSTOPPED,sig=SIGSTOP,EVENT_STOP (128)
strace: next_event: queued pid 2689
strace: next_event: dequeued pid 2689
strace: pid 2689 has TCB_STARTUP, initializing it
strace: [wait(0x80057f) = 2689] WIFSTOPPED,sig=SIGTRAP,EVENT_STOP (128)
strace: next_event: queued pid 2689
strace: next_event: dequeued pid 2689
strace: [wait(0x00127f) = 2689] WIFSTOPPED,sig=SIGCONT
strace: next_event: queued pid 2689
strace: next_event: dequeued pid 2689
strace: [wait(0x00857f) = 2689] WIFSTOPPED,sig=133
strace: next_event: queued pid 2689
strace: next_event: dequeued pid 2689
sstrace: [wait(0x00857f) = 2689] WIFSTOPPED,sig=133
strace: next_event: queued pid 2689
strace: next_event: dequeued pid 2689
strace: [wait(0x00857f) = 2689] WIFSTOPPED,sig=133
strace: next_event: queued pid 2689
strace: next_event: dequeued pid 2689
tracestrace: [wait(0x00857f) = 2689] WIFSTOPPED,sig=133
strace: next_event: queued pid 2689
strace: next_event: dequeued pid 2689
strace: [wait(0x00857f) = 2689] WIFSTOPPED,sig=133
strace: next_event: queued pid 2689
strace: next_event: dequeued pid 2689
:strace: [wait(0x00857f) = 2689] WIFSTOPPED,sig=133
strace: next_event: queued pid 2689
strace: next_event: dequeued pid 2689
strace: [wait(0x00857f) = 2689] WIFSTOPPED,sig=133
```

Strace -c ./merge



```
[10/09/23]seed@VM:~$ strace -c ./merge
Succesfully merged all .txt files data into merged.txt file.
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 60.45    0.000567         141         4           write
 12.69    0.000119          59         2           getdents64
 12.69    0.000119          13         9           openat
  5.44    0.000051           5         9           close
  3.94    0.000037           5         7           read
  3.30    0.000031           3        10           fstat
  1.49    0.000014           4         3           brk
  0.00    0.000000           0         7           mmap
  0.00    0.000000           0         4           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         6           pread64
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         2         1 arch_prctl
------ ----------- ----------- --------- --------- ----------------
100.00    0.000938                    66         2 total
[10/09/23]seed@VM:~$
```

3.Write a program that will categorize all files in the current folder based on their file type. That is all .txt files in one folder called txt, all .bmp files in another folder called bmp etc. The argument to the program is a folder name.

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <dirent.h>

#include <sys/stat.h>

int main(void)

{

DIR *crdir;

char *p1,*p2, ext[100][100], c , filename[50], path[100];

for(int i=0; i<100; i++)
```

```
strcpy(ext[i], "0");
int retn;
struct dirent *dir;
crdir = opendir(".");
if (crdir)
{
while ((dir = readdir(crdir)) != NULL)
{
p1=strtok(dir->d_name,".");
p2=strtok(NULL,".");
if(p2!=NULL)
{
if(strcmp(ext[p2[0]-97], "0") == 0)
strcmp(ext[p2[0]-97], p2);
strcpy(filename, p1);
strcat(filename, ".");
strcat(filename, p2);
mkdir(p2, 0755);
strcpy(path, p2);
strcat(path, "/");
strcat(path, filename);
FILE *fp1 = fopen(path, "w");
FILE *fp2 = fopen(filename, "r");
while((c = fgetc(fp2)) != EOF)
fputc(c, fp1);
}
```

}y

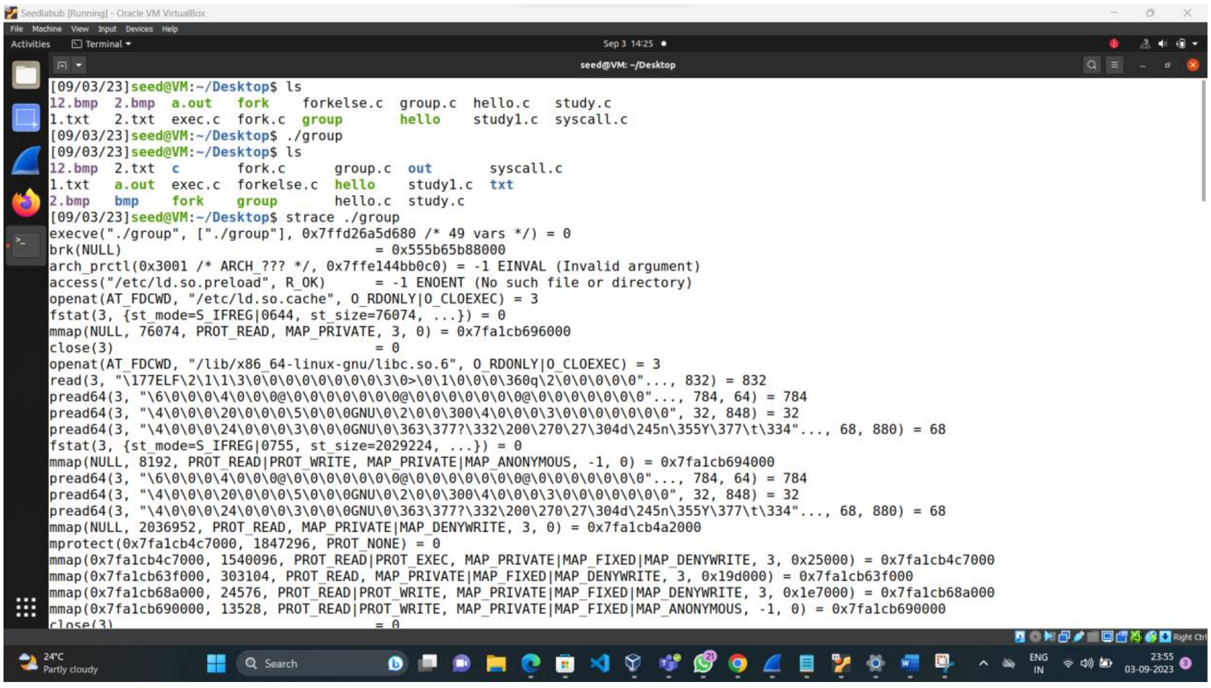
closedir(crdir);

}

return 0;

}

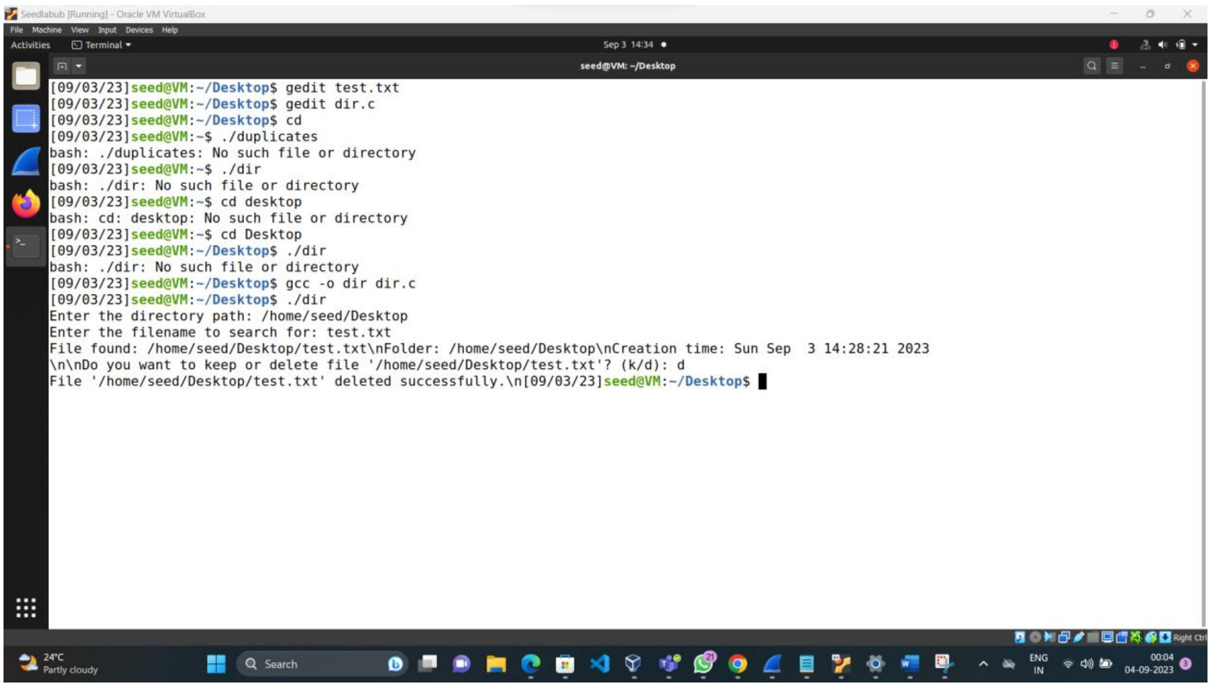
OUTPUT:



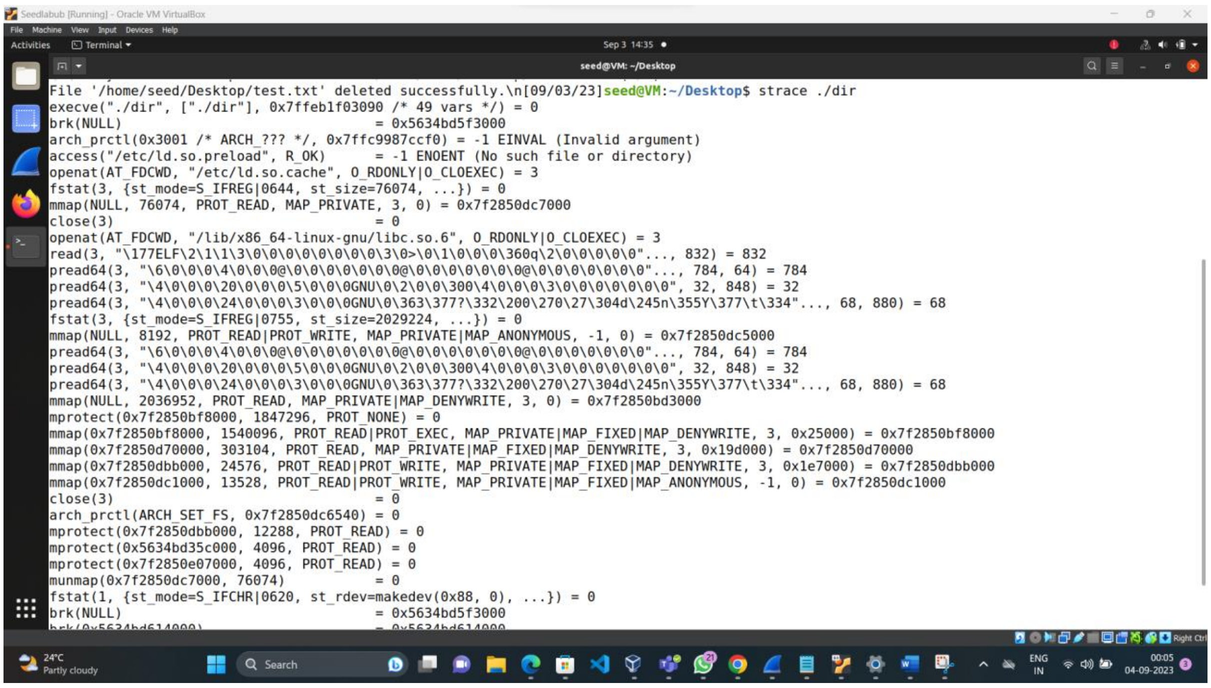
```
[09/03/23]seed@VM:~/Desktop$ ls
12.bmp  2.bmp  a.out   fork     forkelse.c  group.c  hello.c   study.c
1.txt   2.txt  exec.c  fork.c   group       hello    study1.c  syscall.c
[09/03/23]seed@VM:~/Desktop$ ./group
[09/03/23]seed@VM:~/Desktop$ ls
12.bmp  2.txt  c       fork.c      group.c  out       syscall.c
1.txt   a.out  exec.c  forkelse.c  hello    study1.c  txt
2.bmp   bmp    fork    group       hello.c  study.c
[09/03/23]seed@VM:~/Desktop$ strace ./group
execve("./group", ["./group"], 0x7ffd26a5d680 /* 49 vars */) = 0
brk(NULL)                               = 0x555b65b88000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffe144bb0c0) = -1 EINVAL (Invalid argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=76074, ...}) = 0
mmap(NULL, 76074, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fa1cb696000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\2\0\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fa1cb694000
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 68, 880) = 68
mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fa1cb4a2000
mprotect(0x7fa1cb4c7000, 1847296, PROT_NONE) = 0
mmap(0x7fa1cb4c7000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7fa1cb4c7000
mmap(0x7fa1cb63f000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7fa1cb63f000
mmap(0x7fa1cb68a000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7fa1cb68a000
mmap(0x7fa1cb690000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fa1cb690000
close(3)                                = 0
```

4.Given a directory, write a program that will find all files with the same name in the directory and its sub directories. Show their name, which folder they are in and what day they were created. Expand the program to remove all duplicate copies based on user input. That is, ask the user if each one of the files is to be kept or deleted. Based on user input, perform the appropriate action.

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <dirent.h>

#include <sys/stat.h>

#include <time.h>


#define MAX 1000


void find_files(char *basePath, char *filename, int *count, char paths[MAX][MAX]);

void remove_duplicates(char paths[MAX][MAX], int count);


int main()

{

    char filename[MAX];

    char basePath[MAX];

    char paths[MAX][MAX];

    int count = 0;


    printf("Enter the directory path: ");

    scanf("%s", basePath);
    printf("Enter the filename to search for: ");
    scanf("%s", filename);

    find_files(basePath, filename, &count, paths);

    if (count == 0)
        printf("No files found with the name '%s'\\n", filename);
    else
        remove_duplicates(paths, count);

    return 0;
}

void find_files(char *basePath, char *filename, int *count, char paths[MAX][MAX])
{
    char path[MAX];
    struct dirent *dp;
    struct stat buffer;
    DIR *dir = opendir(basePath);

    if (!dir)
        return;

    while ((dp = readdir(dir)) != NULL)
    {
        if (strcmp(dp->d_name, ".") != 0 && strcmp(dp->d_name, "..") != 0)
        {
            strcpy(path, basePath);
            strcat(path, "/");
            strcat(path, dp->d_name);

            if (stat(path, &buffer) == 0 && S_ISDIR(buffer.st_mode))
                find_files(path, filename, count, paths);
            else if (strcmp(dp->d_name, filename) == 0)
            {
                printf("File found: %s\\n", path);
                printf("Folder: %s\\n", basePath);
                printf("Creation time: %s\\n", ctime(&buffer.st_ctime));
                strcpy(paths[*count], path);
                (*count)++;
            }
        }
    }

    closedir(dir);
}

void remove_duplicates(char paths[MAX][MAX], int count)
{
    char ch;
    int i;

    for (i = 0; i < count; i++)
    {
        printf("\\nDo you want to keep or delete file '%s'? (k/d): ", paths[i]);
        scanf(" %c", &ch);
        if (ch == 'd' || ch == 'D')
        {
            if (remove(paths[i]) == 0)
                printf("File '%s' deleted successfully.\\n", paths[i]);
            else
                printf("Unable to delete file '%s'.\\n", paths[i]);
        }
        else
            printf("File '%s' kept.\\n", paths[i]);
    }
}
```


Seedlabub [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Sep 3 14:34
seed@VM: ~/Desktop

```
[09/03/23]seed@VM:~/Desktop$ gedit test.txt
[09/03/23]seed@VM:~/Desktop$ gedit dir.c
[09/03/23]seed@VM:~/Desktop$ cd
[09/03/23]seed@VM:~$ ./duplicates
bash: ./duplicates: No such file or directory
[09/03/23]seed@VM:~$ ./dir
bash: ./dir: No such file or directory
[09/03/23]seed@VM:~$ cd desktop
bash: cd: desktop: No such file or directory
[09/03/23]seed@VM:~$ cd Desktop
[09/03/23]seed@VM:~/Desktop$ ./dir
bash: ./dir: No such file or directory
[09/03/23]seed@VM:~/Desktop$ gcc -o dir dir.c
[09/03/23]seed@VM:~/Desktop$ ./dir
Enter the directory path: /home/seed/Desktop
Enter the filename to search for: test.txt
File found: /home/seed/Desktop/test.txt\nFolder: /home/seed/Desktop\nCreation time: Sun Sep  3 14:28:21 2023
\n\nDo you want to keep or delete file '/home/seed/Desktop/test.txt'? (k/d): d
File '/home/seed/Desktop/test.txt' deleted successfully.\n[09/03/23]seed@VM:~/Desktop$ 
```



```
File '/home/seed/Desktop/test.txt' deleted successfully.\n[09/03/23]seed@VM:~/Desktop$ strace ./dir
execve("./dir", ["./dir"], 0x7ffeb1f03090 /* 49 vars */) = 0
brk(NULL)                               = 0x5634bd5f3000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc9987ccf0) = -1 EINVAL (Invalid argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=76074, ...}) = 0
mmap(NULL, 76074, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f2850dc7000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\2\0\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f2850dc5000
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 68, 880) = 68
mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f2850bd3000
mprotect(0x7f2850bf8000, 1847296, PROT_NONE) = 0
mmap(0x7f2850bf8000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7f2850bf8000
mmap(0x7f2850d70000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7f2850d70000
mmap(0x7f2850dbb000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f2850dbb000
mmap(0x7f2850dc1000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f2850dc1000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7f2850dc6540) = 0
mprotect(0x7f2850dbb000, 12288, PROT_READ) = 0
mprotect(0x5634bd35c000, 4096, PROT_READ) = 0
mprotect(0x7f2850e07000, 4096, PROT_READ) = 0
munmap(0x7f2850dc7000, 76074)           = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
brk(NULL)                               = 0x5634bd5f3000
```