**LAPORAN PRAKTIKUM**

**SISTEM OPERASI**

**MODUL 9**



**DISUSUN OLEH:**

|  |  |
| --- | --- |
| **NIM** | **L200220277** |
| **NAMA** | **MHD. FARHAN LUBIS** |
| **KELAS** | **F** |

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS KOMUNIKASI DAN INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2023**

# DAFTAR ISI

[**DAFTAR ISI 2**](#_Toc152067316)

[**LANGKAH KERJA 4**](#_Toc152067317)

[**Membuat sebuah file dan menuliskan data di dalamnya. 4**](#_Toc152067318)

[**1. Deklarasi variabel untuk buffer ‘buf’ untuk menyimpan 100 byte data. 5**](#_Toc152067319)

[**2. Gunakan nama file yang diberikan sebagai argumen pada command-line 5**](#_Toc152067320)

[**3. Buat sebuah file baru dengan nama file seperti pada (2) menggunakan fungsi system-call open() dengan opsi O\_ CREAT dan O\_TRUNC. 5**](#_Toc152067321)

[**4. Periksa ‘file-descriptor’, jika pembuatan file tidak berhasil, selanjutnya berhenti (stop) dan keluar dari program. 5**](#_Toc152067322)

[**5. Baca input dari console sampai use menekan tombol Ctrl+D. Membaca 100 byte dari data yang dimasukan mellaui console dan menyimpannya ke dalam variabel ‘buf’ menggunakan perintah system-call read(). Memindahkan isi variabel ‘buf’ ke dalam file menggunakanperintah ‘write’. 5**](#_Toc152067323)

[**6. Menutup file dengan menggunakan fungsi ‘close’ 5**](#_Toc152067324)

[**7. Stop 5**](#_Toc152067325)

[**Kode program: 5**](#_Toc152067326)

[**Terminal ketika program dicompile & dijalankan: 6**](#_Toc152067327)

[**Membaca sebuah file dan menampilkan isinya di layar. 6**](#_Toc152067328)

[**1. Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data. 7**](#_Toc152067329)

[**2. Gunakan nama-file sesuai dengan argumen yang diberikan dalam perintah command-line. 7**](#_Toc152067330)

[**3. Buka file untuk dibaca menggunakan perintah ‘open’ dengan opsi O\_RDONLY. 7**](#_Toc152067331)

[**4. Periksa isi file-descriptor, Jika file tidak ada maka program berhenti, stop 7**](#_Toc152067332)

[**5. Baca isi file per 100 byte data menggunakan perintah ‘read’ sampai ketemu dengan tanda akhir file, ‘END-OFFILE’. 7**](#_Toc152067333)

[**6. Tutup file menggunakan perintah ‘close’ 7**](#_Toc152067334)

[**7. Stop 7**](#_Toc152067335)

[**Kode program: 7**](#_Toc152067336)

[**Terminal ketika program dicompile & dijalankan: 8**](#_Toc152067337)

[**Menambah isi file. 8**](#_Toc152067338)

[**a. Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data. 9**](#_Toc152067339)

[**b. Gunakan nama-file sesuai dengan argumen yang diberikan dalam perintah command-line. 9**](#_Toc152067340)

[**c. Buka file di atas dengan menggunakan perintah open dengan opsi O\_APPEND. 9**](#_Toc152067341)

[**d. Periksa file-descriptor, jika nilainya berupa angka negatif, stop program 9**](#_Toc152067342)

[**e. Baca input user dari console sampai user menekan tombol Ctrl + D. Jika user sudah menekan tombol Ctrl + D, baca 100 byte dari console dan simpan ke dalam variabel ‘buf’ menggunakan perintah read. Selanjutnya tuliskan isi variabel ‘buf’ ke dalam file mengunakan perintah ‘write’. 9**](#_Toc152067343)

[**f. Tutup file dengan menggunakan fungsi ‘close’. 9**](#_Toc152067344)

[**Kode program: 9**](#_Toc152067345)

[**Terminal ketika program dicompile & dijalankan: 10**](#_Toc152067346)

# LANGKAH KERJA

## Membuat sebuah file dan menuliskan data di dalamnya.

Membuat program dengan algoritma sebagai berikut (Contoh program diberikan pada bagian berikutnya):

### Deklarasi variabel untuk buffer ‘buf’ untuk menyimpan 100 byte data.

### Gunakan nama file yang diberikan sebagai argumen pada command-line

### Buat sebuah file baru dengan nama file seperti pada (2) menggunakan fungsi system-call open() dengan opsi O\_ CREAT dan O\_TRUNC.

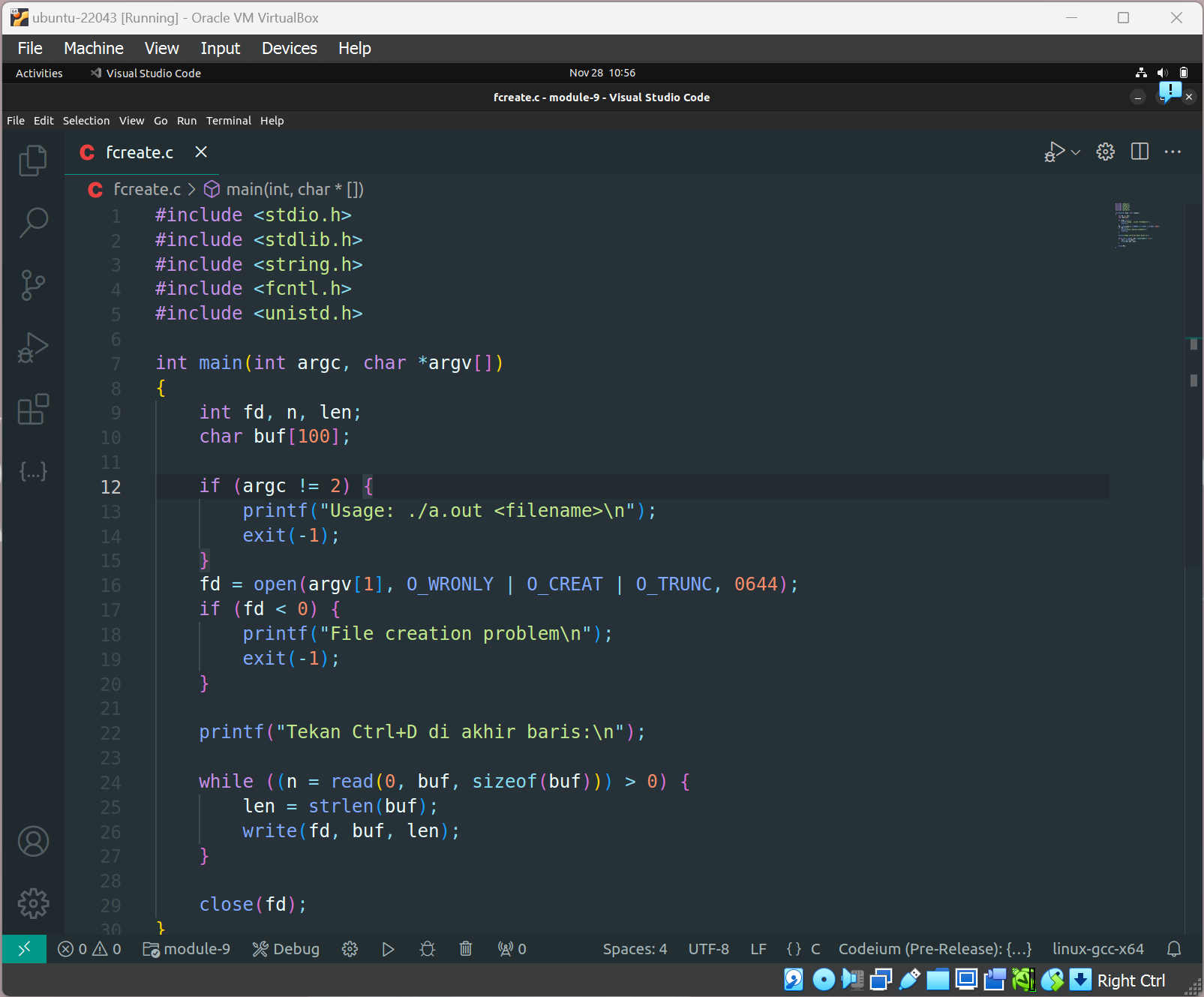
### Periksa ‘file-descriptor’, jika pembuatan file tidak berhasil, selanjutnya berhenti (stop) dan keluar dari program.

### Baca input dari console sampai use menekan tombol Ctrl+D. Membaca 100 byte dari data yang dimasukan mellaui console dan menyimpannya ke dalam variabel ‘buf’ menggunakan perintah system-call read(). Memindahkan isi variabel ‘buf’ ke dalam file menggunakan perintah ‘write’.

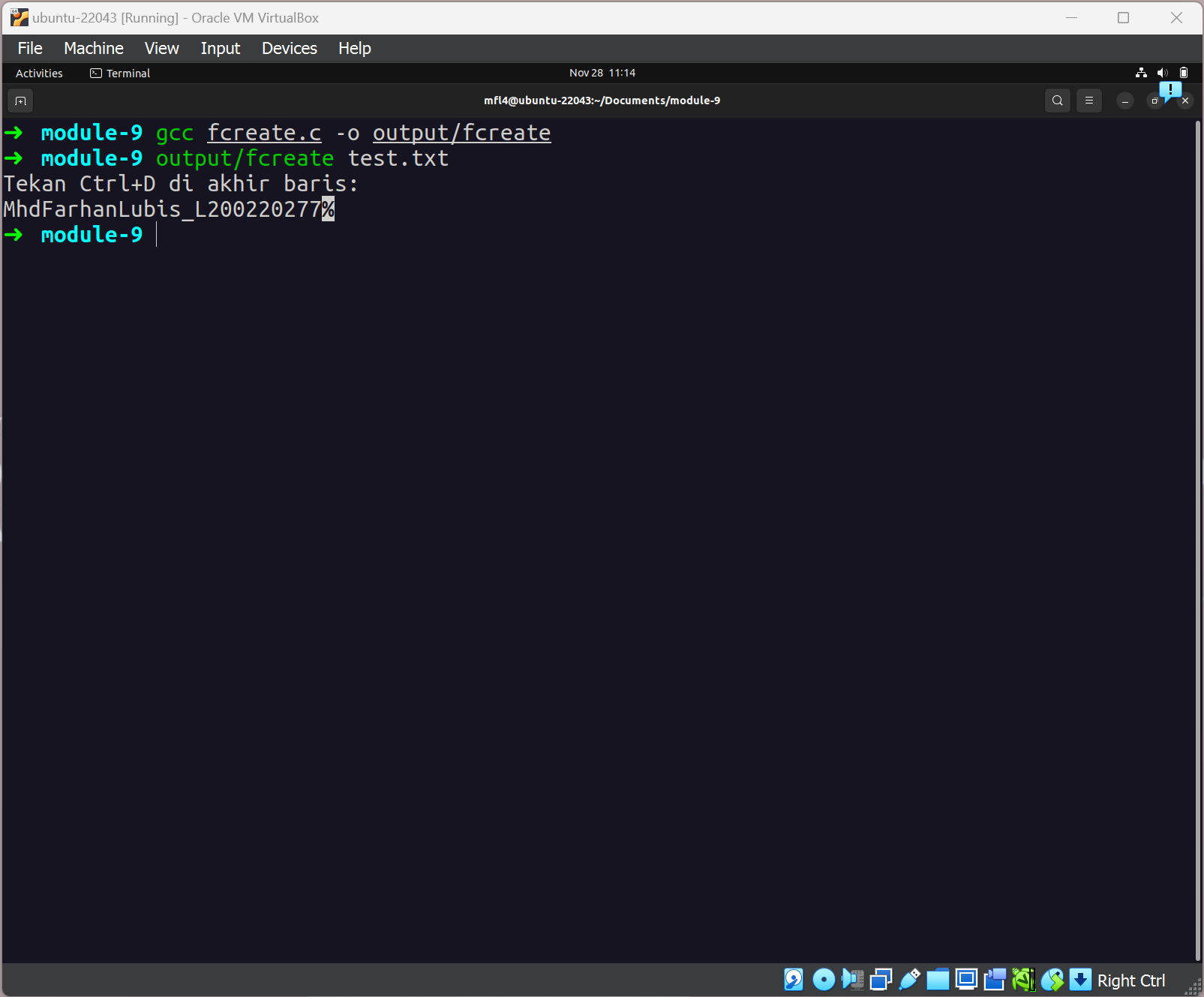
### Menutup file dengan menggunakan fungsi ‘close’

### Stop

### Kode program:



### Terminal ketika program dicompile & dijalankan:



Compilasi program dengan perintah $gcc -o fcreate fcreate.c , selanjutnya jika proses compilasi berhasil jalankan program dengan memanggil nama file $ ./fcreate test.txt, ketikan text yang akan disimpan dalam file, tekan Ctrl + D untuk menandai baris baru, dan tekan Ctrl + C jika sudah selesai.

## Membaca sebuah file dan menampilkan isinya di layar.

Membuat kode program dengan algorithm sebagai barikut:

### Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data.

### Gunakan nama-file sesuai dengan argumen yang diberikan dalam perintah command-line.

### Buka file untuk dibaca menggunakan perintah ‘open’ dengan opsi O\_RDONLY.

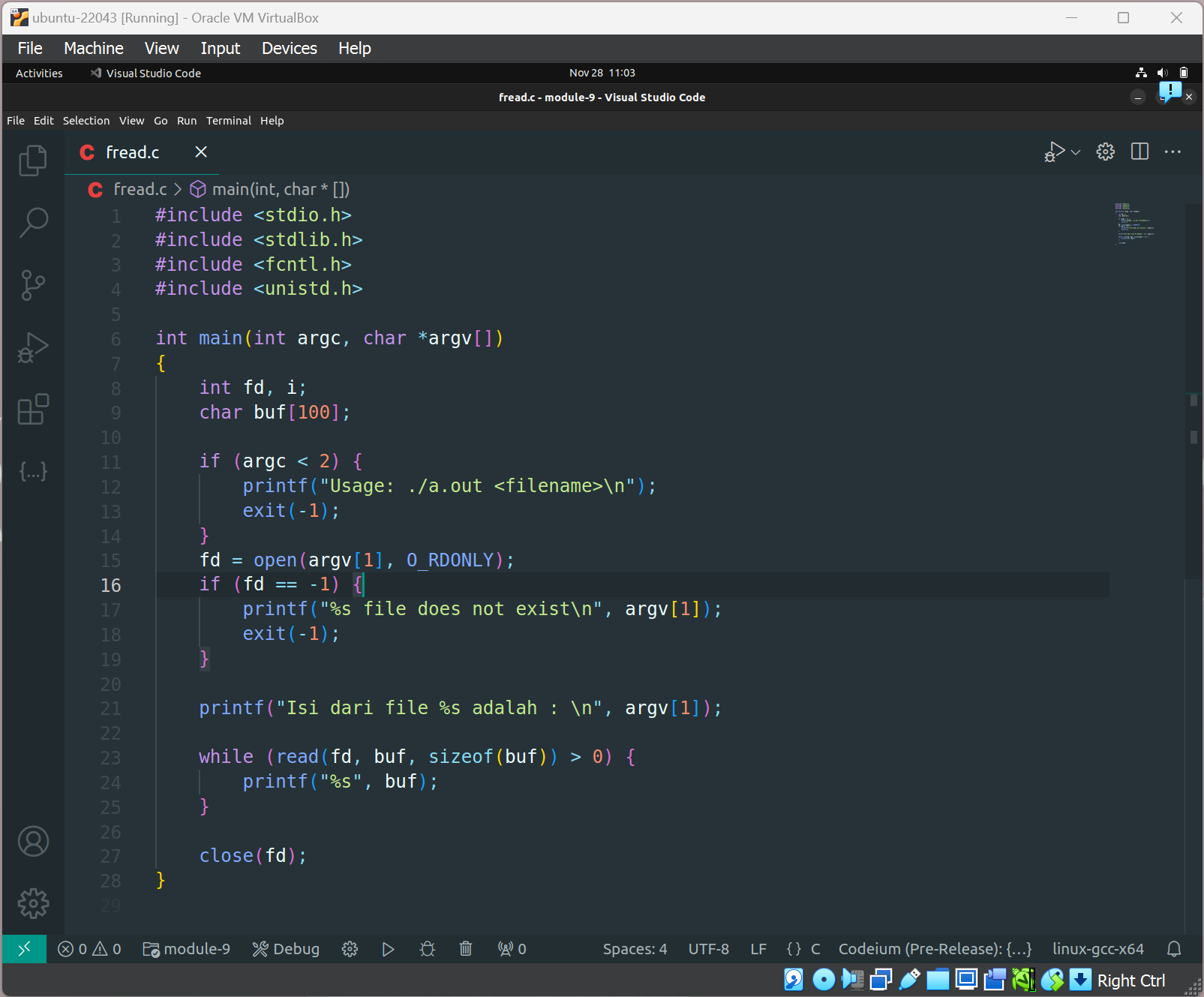
### Periksa isi file-descriptor, Jika file tidak ada maka program berhenti, stop

### Baca isi file per 100 byte data menggunakan perintah ‘read’ sampai ketemu dengan tanda akhir file, ‘END-OFFILE’.

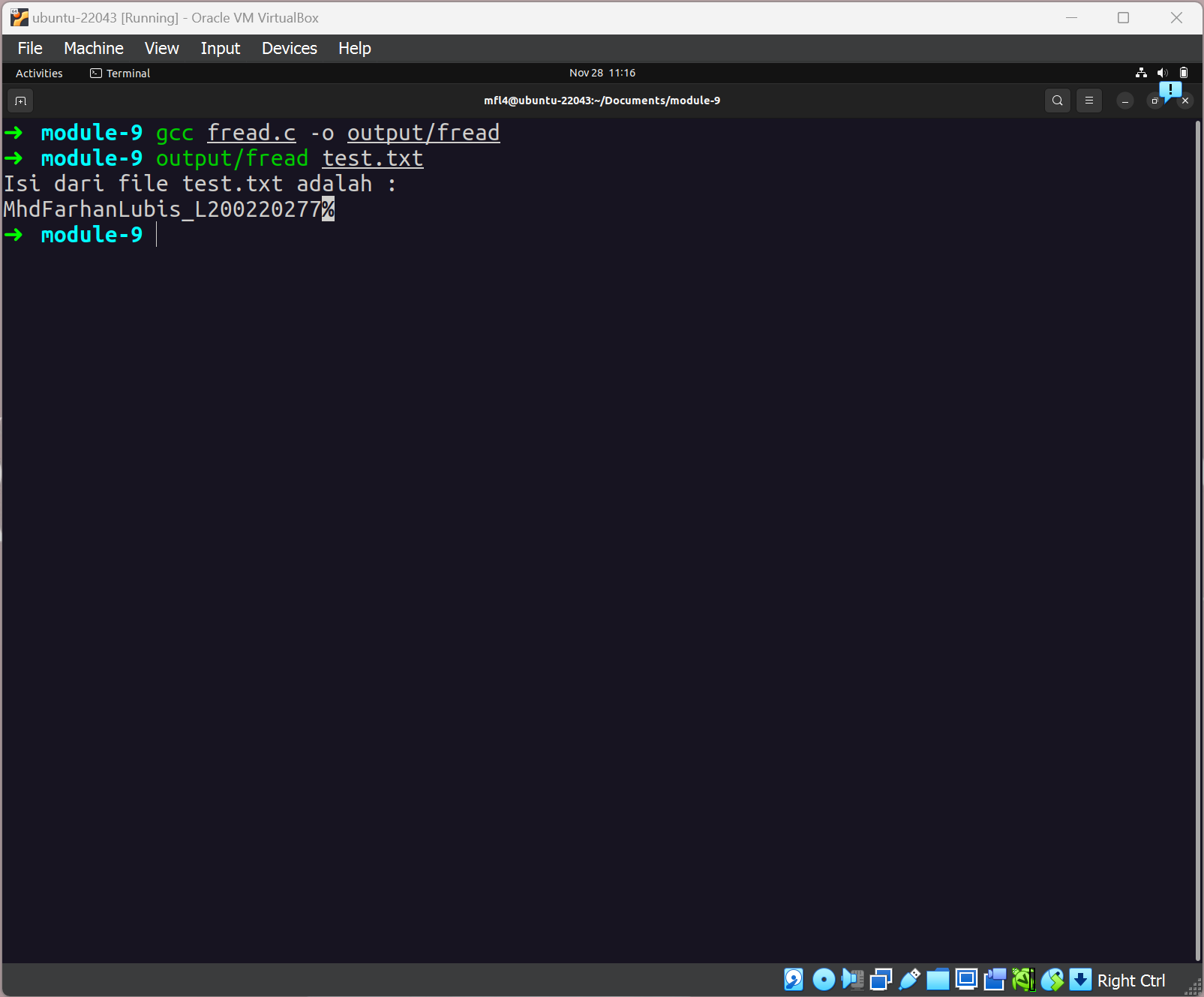
### Tutup file menggunakan perintah ‘close’

### Stop

### Kode program:



### Terminal ketika program dicompile & dijalankan:



## Menambah isi file.

Membuat kode program dengan algorithm sebagai barikut:

### Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data.

### Gunakan nama-file sesuai dengan argumen yang diberikan dalam perintah command-line.

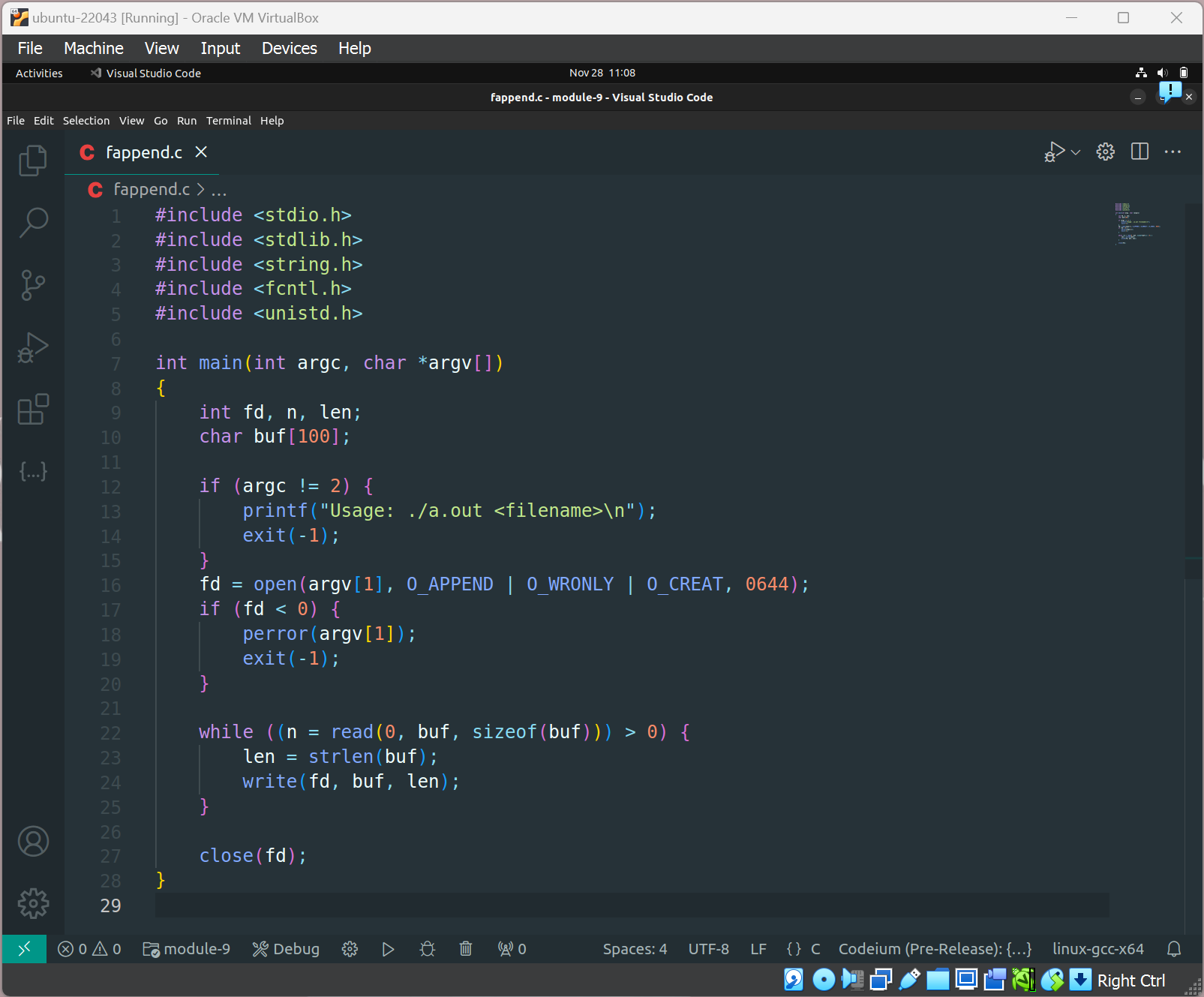
### Buka file di atas dengan menggunakan perintah open dengan opsi O\_APPEND.

### Periksa file-descriptor, jika nilainya berupa angka negatif, stop program

### Baca input user dari console sampai user menekan tombol Ctrl + D. Jika user sudah menekan tombol Ctrl + D, baca 100 byte dari console dan simpan ke dalam variabel ‘buf’ menggunakan perintah read. Selanjutnya tuliskan isi variabel ‘buf’ ke dalam file menggunakan perintah ‘write’.

### Tutup file dengan menggunakan fungsi ‘close’.

### Kode program:



### Terminal ketika program dicompile & dijalankan:

