Roy Sumurung Sigalingging
11323029
D3-TI

MEMORI VIRTUAL

1) Tujuan
- Menjelaskan manfaat sistem memori
- Membahas prinsip model himpunan kerja
- untuk menjelajahi bagaimana memori korret dikelolah.
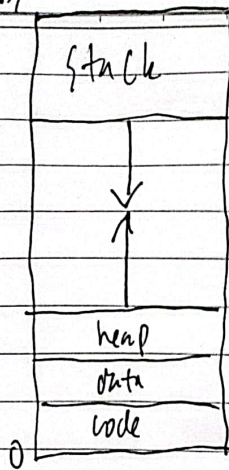
2) Latar belakang
- Kode harus ada di memori untuk dieksekusi, tetapi keseluruhan program jarang digunakan.
- Kesalahan kode, rutinitas yang tdk biasa, struktur data yang besar.
- Seluruh kode program tdk diperlukan secara bersamaan.
- Program tdk dibatasi oleh batasan memori fisik.
- Setiap program membutuhkan lebih sedikit memori saat dijalankan → lebih banyak program dijalankan secara bersamaan.
- lebih sedikit I/O yang diperlukan utk memuat program kedlm memori → setiap program pengguna berjalan lebih cepat.
- memori maya → pemisahan memori logis pengguna dari memori fisik
- ruang alamat virtual → pandangan logis ttg bagaimana proses disimpan dlm memori
- memori virtual dapat diimplementasikan melalui :
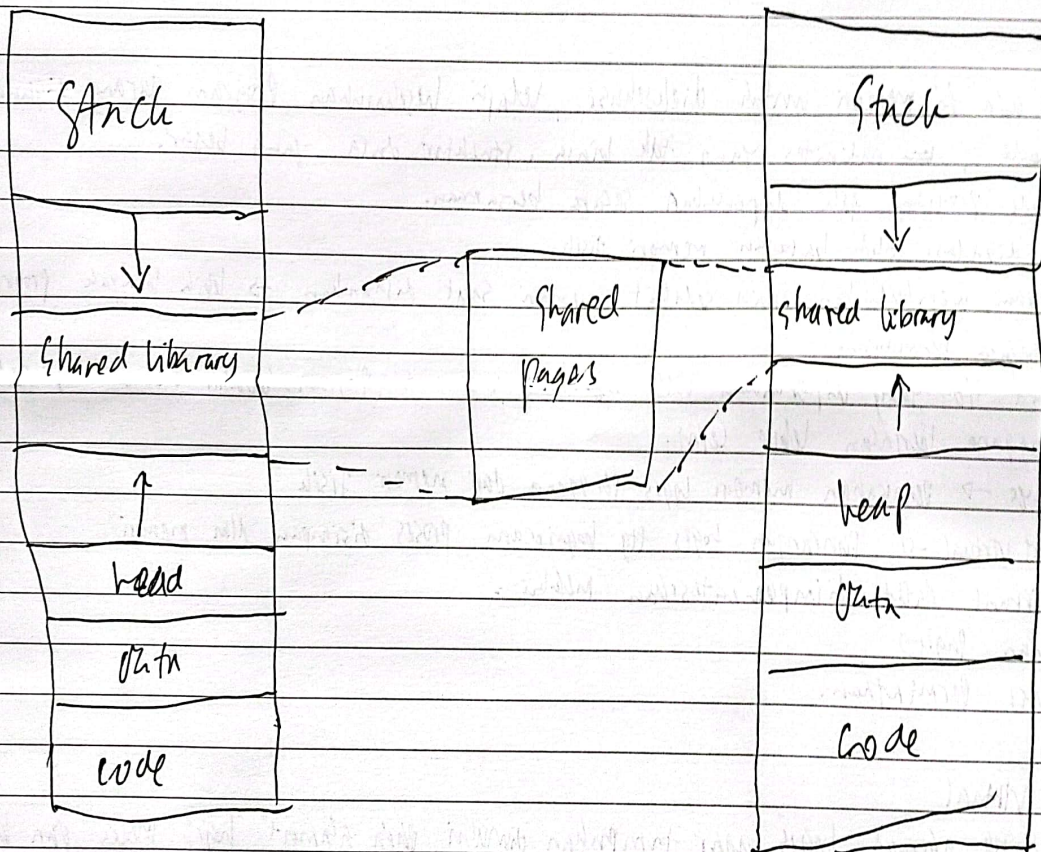 - Permintaan paging
  - segmentasi permintaan.

3) ruang alamat virtual
- merancang ruang alamat logis agar tumpukan dimulai pada alamat logis maks dan bertambah "turun" sementara heap bertambah "naik".
- memaksimalkan penggunaan ruang alamat.
- ruang alamat yg tdk dipakai diantara keduanya adlh lubang.
- memungkinkan jarang ruang alamat dgn lubang tersisa untuk pertumbuhan, perpustakaan yg terhubung secara dinamis
- pustaka sistem dibagikan melalui pemetaan ke ruang alamat virtual
- memori bersama dgn memetakan halaman baca-tulis kedlm ruang alamat virtual.

MAP


```
┌──────────┐
│  Stack   │
│          │
│    ↓     │
│    ↑     │
│          │
│   heap   │
│   data   │
│   code   │
0└──────────┘
```

5) Perpustakaan bersama menggunakan memori virtual


```
┌──────────┐                    ┌──────────┐
│  Stack   │                    │  Stack   │
│          │                    │          │
│    ↓     │                    │    ↓     │
│          │   ┌─────────┐      │          │
│          │   │ Shared  │      │ shared library │
│ Shared library │ pages │      │          │
│          │   └─────────┘      │    ↑     │
│    ↑     │                    │   heap   │
│   head   │                    │   data   │
│   data   │                    │          │
│   code   │                    │   Code   │
└──────────┘                    └──────────┘
```

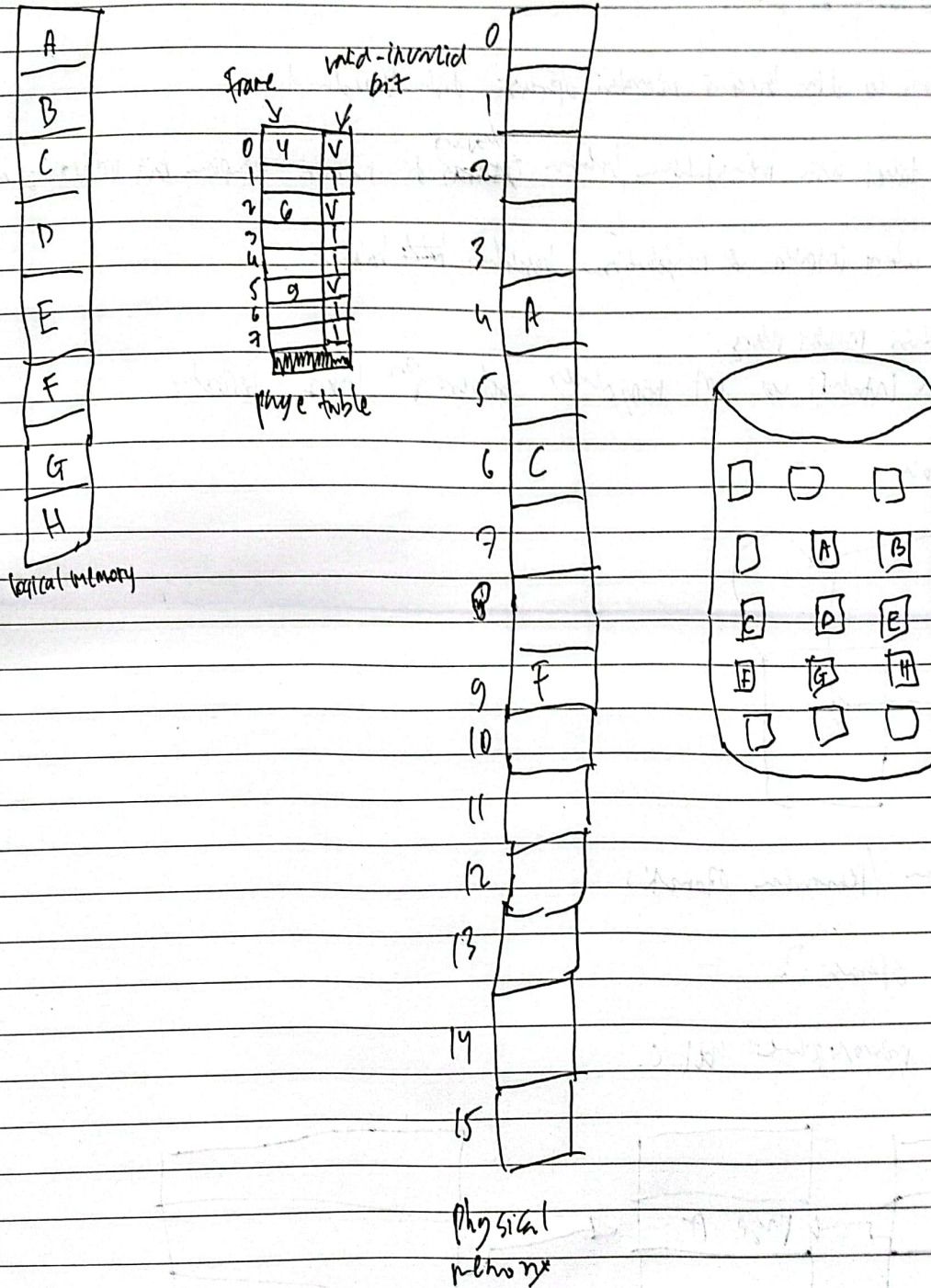Permintaan Paging

- tdk membawa seluruh proses ke dlm memori pd waktu tertentu.
- bawa halaman ke dlm memori hanya saat diperlukan.
- mirip dgn sistem paging dgn swaping.
- halaman diperlukan ⇒ referensi utk itu.
- penaksir yg malas ⇒ dgn pernah penaksir halaman. ke dlm memori menjadi belum diperlukan
- swapper yg berhubungan dgn halaman adalah apager.

SiDU

6) konsep dasar
* dgn swapping, Pager mengambil halaman mana yg akan digunakan sebelum menukarnya lagi.
* Pager hanya membawa halaman ke hoffer ke memori.

6) tabel halaman ketika beberapa hal telah ada di memori utama.

A
B
C
D
E
F
G
H

logical memory

frame   valid-invalid bit

0	4	V
1		i
2	6	V
3		i
4		i
5	9	V
6		i
7		i

page table

0
1
2
3
4 A
5
6 C
7
8
9 F
10
11
12
13
14
15

Physical memory

a) kesalahan halaman

2) Sistem ~~operasi~~ operasi melihat tabel ~~ini~~ lain utk pemeriksaan :
  * Referensi tdk valid => menggugurkan
  * harga tdk ada dlm ingatan.

2) kesalahan bingkai bebas.

3) tukar halaman ke dlm bingkai memulai operasi disk tergantung

4) saat ulang tabel utk menunjukkan halaman sekarang di memori tetapkan bit validasi = ny

  ☞ mulai ulang instruksi yg menyebabkan kesalahan halaman.

5) instruksi mulai ulang.
  * pertimbangkan instruksi yg dpt mengakses beberapa lokasi berbeda
  ↳ deralam blok

  z lokasi kesalahan (kemungkinan otomatis)
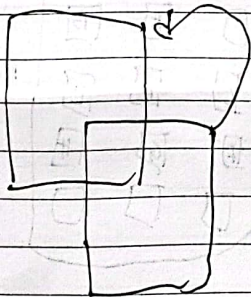
  ₽ mulai ulang operasi ?

  5) sebelum proses 1 memodifikasi hal C.

  prosesor 1   physical   prosesor 2
                      memory
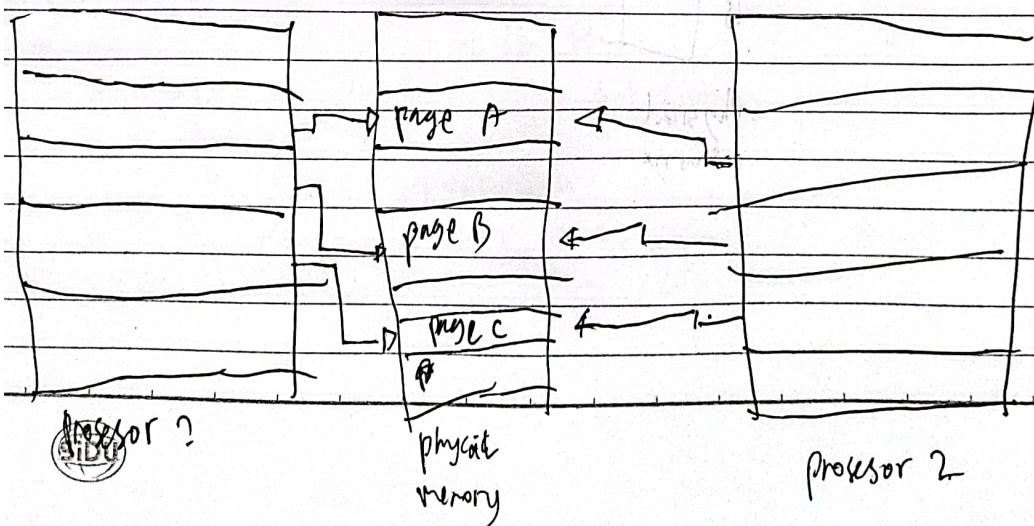
Roy Sigalingging
11323024
D3 T1

Antar Muka
Sistem File

> konsep berkas
- jenis ruang alamat logis yg
- berdekatan : - data
  - numerik
  - karakter
  - biner
  - program
- konten ditentukan oleh pembuat file

> Atribut berkas
- nama
- pengidentifikasi
- lokasi file di perangkat dimana file saat ini
- perlindungan
- waktu, tanggal, dan identifikasi pengguna
- informasi ttg file disimpan dlm struktur direktori, yg disimpan di disk
- banyak variasi, termasuk atribut file yg diperluas seperti file the.
  Checksum
- informasi disimpan dlm struktur direktori.

> operasi berkas.
- file adlh tipe data abstrak
- buat
- menulis penunjuk lokasi·
- pembaca penunjuk
- lokasi reposisi dlm file
- menghapus
- menobati
- terbuka
- tertutup

6) buka file
  * tabel file terbuka
  * petunjuk berkas
  * jumlah pembukaan file
  * lokasi disk file
  * hak akses

6) lock penguncian file
  * mirip dgn kunci pembaca-penulis
  * kunci bersama mirip dgn kunci pembaca.
  * kunci eksklusif. mirip dgn kunci penulis.
  * ~~akse~~ meredial akses ke file
  * wajib
  * penasehat

6) struktur berkas
  * tidak ada ~~as~~ urutan byte, byte
  * struktur rekaman sederhana.
  * baris
  * panjang tetap
  * " variabel.
  * struktur komplek
  * dokumen yg diformat
  * file beban yg dpt direlokasi

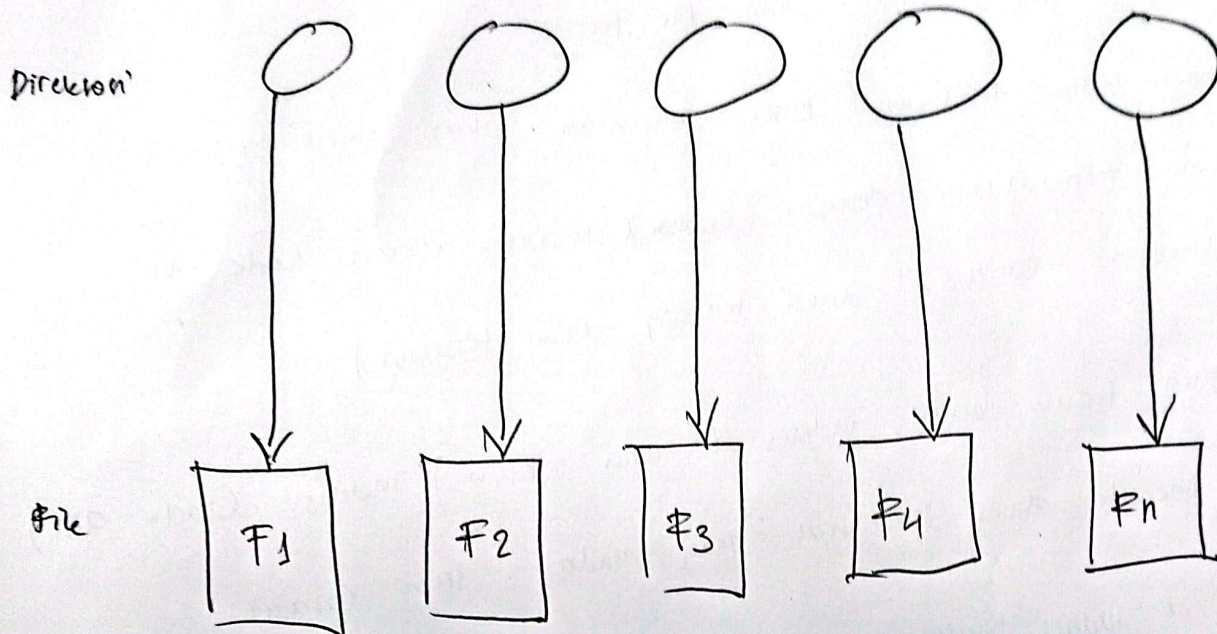6) file akses berurutan
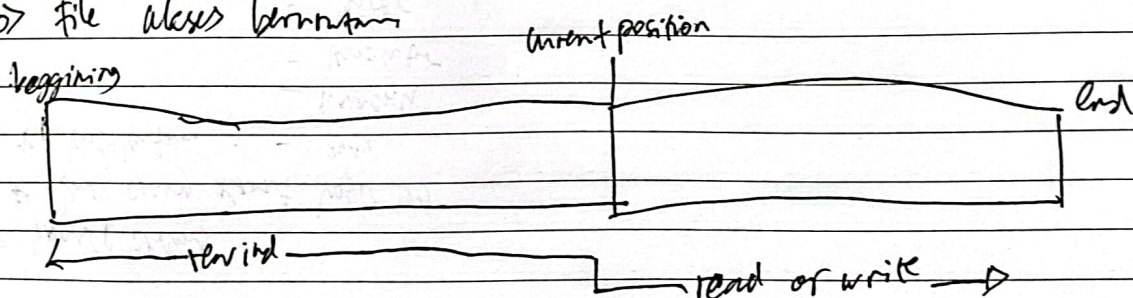
7) ~~struktur~~ cth file indeks dan relatif

last name   logical record

adams	
Arthr	
asher	
:	
Smith	

Smith, John | social-security | age

relative file

– kumpulkan node yang berisi informasi tentang semua file

Direktori

file

F1   F2   F3   F4   Fn

Struktur direktori dan file berada di disk.

Jenis Sistem File

* Peninbangkan folder punya

  * tmpFS – FS kolam² berbasis memori untuk I/O sementara yang cepat

  * Obss – aurar mulca ke memori kernel untuk mendapatkan simbol kernel untuk debugging.

  * CtFS – sistem file kontrak untuk mengelola daemon.

  * procFS – auramulca kernel untuk memproses.

Metode Akses lainnya

* Dapat dibangun diatas metode dasar umum membaca

* Penciptaan sebuah indeks untuk file tersebut

* simpan indeks di memori untuk penemuan lokasi data yang
  akan dioperasikan dengan cepat (penimbangan kode upc
  ditambah catatan data tentang item tersebut)

* Jika terlalu besar, indeks (dalam memori) indeks (pada disk)

* Metode akses sekuensial yang diindeks IBM (isam)

    * Indeks master kecil, menunjuk ke blok disk indeks sekunder

    * File disimpan diurutkan pada kunci yang diurutkan.

    * semua dilakukan oleh OS.

* Sistem operasi Vrms menyediakan file indeks dan relatif
  sebagai contoh lain (

* kumpulan node yang berisi informasi tentang semua file.

* disk dapat dibagi bagi menjadi partisi

* disk atau partisi bisa serangan terlindungi dari kegagalan

* Berisi sistem file yang ditemui sebagai direktori