



MANAJEMEN INPUT OUTPUT

Core Jurusan Teknik Informatika
Kode MK/SKS : TIF 01 0 4 10/ 4

Subsistem I/O pada Kernel

- Kernel banyak menyediakan service yang berhubungan dengan I/O:
I/O scheduling, Buffering, Caching, Spooling, Error handling



- I/O scheduling:

Untuk menjadwalkan sebuah set permintaan I/O, harus menentukan urutan yang baik untuk mengeksekusi permintaan tersebut.



- **Buffering:**

Area memori yang menyimpan data ketika data sedang dipindahkan antara dua device atau antara device dan aplikasi.



- **Caching:**

Daerah memori yang cepat , berisikan copy-an data.



- Spooling:

Buffer yang menyimpan output untuk sebuah device, seperti printer, yang tidak dapat menerima interleaved data streams



- **Error Handling:**

Sistem operasi yang menggunakan protected memory dapat menjaga banyak kemungkinan error akibat hardware maupun aplikasi.



Transformasi I/O ke operasi hardware

- Dapat mengenalkan device dan driver baru ke komputer tanpa harus meng-compile ulang kernelnya.



Transformasi I/O ke operasi hardware

- Sebagai fakta, ada beberapa sistem operasi yang mampu me-load device drivers yang diinginkan.



Transformasi I/O ke operasi hardware

- Pada saat boot, sistem mula-mula meminta bus peranti keras untuk menentukan device apa yang tersedia, lalu sistem me-load ke dalam driver yang sesuai





STUDI KASUS

Core Jurusan Teknik Informatika
Kode MK/SKS : TIF 01 0 4 10/ 4

Manajemen Memori

- MS-DOS:
 - Conventional Memori
 - Upper memori
 - High Memor
 - Extended dan Expanded Memori



Manajemen Memori

- Windows NT:
Manajer memori dirancang untuk beroperasi di atas berbagai platform dan menggunakan page yang berukuran 4 – 64 kb.



- Platform Intel, Power PC, dan MIPS memiliki 4096 byte per page nya. Sedangkan DEC Alpha memiliki 8192 byte per page



Manajemen Memori

- Linux:
Mengimplementasikan sistem virtual memori demand page. Proses mempunyai memori virtual yang besar (4 gigabyte). Pada virtual memori dilakukan transfer page antara disk dan memori fisik.

Penggunaan Memori Virtual

- Windows NT:
Mengimplementasikan memori virtual dengan menggunakan demand paging melalui clustering.



- Clustering menangani page fault dengan menambahkan tidak hanya page yang terkena fault, tetapi juga beberapa page yang ada dekat page tersebut.



Penggunaan Memori Virtual

- Linux:
Menggunakan variasi dari algoritma clock. Thread dari kernel Linux (kswapd) akan dijalankan secara periodik atau dipanggil saat penggunaan memori sudah berlebihan