|  |
| --- |
| **LAPORAN PRAKTIKUM SISTEM OPERASI**  **MANAGEMENT MEMORY** |
|  |
| **Agus Pranata Marpaung**  **13323033**  **DIII TEKNOLOGI KOMPUTER** |
| **INSTITUT TEKNOLOGI DEL**  **FAKULTAS VOKASI** |

**Judul Praktikum**

|  |  |  |
| --- | --- | --- |
| **Minggu/Sesi** | : | XII/2 dan 3 |
| **Kode Mata Kuliah** | : | 1031202/1041202 |
| **Nama Mata Kuliah** | : | SISTEM OPERASI |
| **Setoran** | : | Panduan ini dibuat untuk mengarahkan mahasiswa memahami mengenai Management Memory dalam upaya membantu mereka melaksanakan Pembelajaran Jarak Jauh (PJJ). Panduan ini adalah bagian pertama untuk topik Manajemen Memori. |
| **Batas Waktu Setoran** | : | *24 April 2024 jam 17:00* |
| **Tujuan** | : | 1. Mampu menjelaskan tujuan dari manajemen memori. 2. Mampu menjelaskan mengenai relocation, protection, sharing, logical organization dan physical organization dalam melakukan manajemen memori. 3. Mampu menjelaskan jenis-jenis teknik partisi pada memori seperti fixed partitioning, dynamic partitioning, simple paging dan simple segmentation. 4. Mampu menerapkan algoritma penempatan pada dynamic partitioning. |

**Petunjuk**

1. Tugas ini dikerjakan secara individu.
2. Mencontoh pekerjaan dari orang lain akan dianggap plagiarisme dan anda akan ditindak sesuai dengan sanksi akademik yang berlaku di IT Del atau sesuai dengan kebijakan saya dengan memberikan nilai 0.
3. Jawaban diketikkan dalam bentuk laporan mengikuti template yang telah disediakan di- ecourse dan setiap soal harus ditulis secara berurutan.
4. Keterlambatan menyerahkan laporan tidak ditolerir dengan alasan apapun. Oleh karena itu, laporan harus dikumpul tepat waktu.

**Referensi**

* Stalling William, Operating Systems: Internal and Design Principles, 7th edition, Chapter 7, Prentice Hall, 2012.
* A. Silberschatz, P.B. Galvin, and G. Gagne, Operating System Concepts, 9th edition, Chapter 9, John Wiley & Sons, Inc., 2013.

**Management Memory**

**Teori**

1. Jelaskan mengapa manajemen memori diperlukan?

**Jawab:**

Karena manajemen memori sangat penting karena memastikan penggunaan memori yang efisien dengan memberikan ruang memori yang tepat untuk setiap proses atau aplikasi. Selain itu, manajemen memori melindungi data dan kode program dari gangguan dan akses yang tidak sah, dengan memisahkan proses dari proses lainnya. Jika diperlukan, memori dapat dibagi dan dialokasikan kepada berbagai proses dengan manajemen memori. Mengelola memori virtual adalah fungsi tambahan dari manajemen memori, yang memungkinkan sistem menggunakan lebih banyak memori daripada yang sebenarnya tersedia dalam bentuk RAM.

1. Jelaskan fungsi manajemen memori!

**Jawab:**

Berikut beberapa fungsi manajemen memori:

1. **Alokasi Memori**

Fungsi dari alokasi memori yaitu mengalokasikan ruang memori yang tepat untuk setiap proses atau aplikasi dan memastikan penggunaan memori yang efektif adalah tugas manajemen memori.

1. **Perlindungan dan Isolasi**

Fungsi dari perlindungan dan Isolasi yaitu dengan Dengan memisahkan setiap proses dari proses lainnya, manajemen memori melindungi data dan kode program dari gangguan dan akses yang tidak sah.

1. **Pembagian Memori**

Fungsi dari pembagian memori yaitu memori dapat dibagi dan dialokasikan kepada berbagai proses melalui manajemen memori.

1. **Pengelolaan Memori Virtual**

Fungsi dari pengelolaan Memori Virtual yaitu memori virtual dikelola oleh manajemen memori, yang memungkinkan sistem menggunakan lebih banyak memori daripada yang sebenarnya tersedia dalam bentuk RAM.

1. **Penggantian dan Pemindahan**

Fungsi dari penggantian dan pemindahan yaitu manajemen memori bertanggung jawab untuk mengganti blok memori yang tidak lagi diperlukan dan memindahkan blok memori ke tempat yang berbeda.

1. Jelaskan sifat dari manajemen memori pada sistem operasi yang mendukung *monoprogromming* dan *multiprogramming*!

**Jawab:**

1. ***Monoprogramming***

Manajemen memori menjadi lebih mudah dalam sistem operasi yang mendukung monoprogramming karena hanya ada satu program yang dapat dijalankan pada satu waktu. Semua memori dapat dialokasikan untuk program tersebut, kecuali memori yang digunakan sistem operasi itu sendiri. Karena hanya ada satu program yang berjalan, tidak perlu melindungi memori atau membagi memori.

1. ***Multiprogramming***

Sebuah program dapat dijalankan secara bersamaan di sistem operasi yang mendukung multiprogramming. Akibatnya, manajemen memori menjadi lebih rumit. Semua program yang berjalan harus memiliki memori terpisah, dan setiap program harus tidak dapat diakses oleh program lain. Saat program dimulai dan dihentikan, manajemen memori juga harus dapat mengalokasikan dan dealokasikan memori untuk program ini.

1. Jelaskan definisi dari:
   1. *Frame*
   2. *Page*
   3. *Segment*

**Jawab:**

1. ***Frame***

Frame merujuk pada blok yang berukuran tetap dalam ruang memori fisik atau blok penyimpanan pusat.

1. ***Page***

Page (halaman memori) ialah blok kontigu yang beukuran tetap dari memori virtual, yang dijelaskan oleh satu entri pada tabel halaman.

1. ***Segment***

Segment ialah blok kontinu memori dengan ukuran tertentu. Untuk mengakses memori dalam setiap segmen, diperlukan offset. Salah satu ide dari segmentasi adalah memperbesar memori yang hanya memiliki register 16-bit.

1. Jelaskan definisi dari:
   1. *Register base*
   2. *Register limit*
   3. *Symbolic address*
   4. *Relocatable address*
   5. *Absolute address*
   6. *Logical address*
   7. *Physical address*

**Jawab:**

1. ***Register base***

Register base merupakan suatu register yang digunakan pada sistem komputer untuk menyimpan Alamat dasar dari blok memori.

1. ***Register limit***

Register limit merupakan suatu register yang digunakan untuk menyimpan ukuran dari blok memori.

1. ***Symbolic address***

Symbolic address merupakan suatu skema penamaan Alamat Dimana referensi ke Alamat dibuat dengan simbol yang mudah diingat dan memiliki hubungan dengan data yang diharapkan berada di Alamat tersebut.

1. ***Relocatable address***

Relocatable address merupakan suatu alamat yang dapat dipindahkan atau diubah. Pada manajemen memori, alamat simbolis dari program sumber di disk diikat oleh compiler ke alamat yang dipindahkan (yang merupakan alamat instruksi relative satu sama lain).

1. ***Absolute address***

Absolute address merupakan suatu alamat memori yang tepat digunakan oleh perangkat keras atau perangkat lunak.

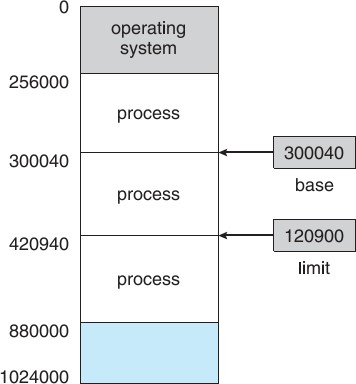
1. ***Logical address***

Logical address merupakan alamat yang dihasilkan oleh CPU selama mengeksekusi program. Alamat ini adalah alamat yang dilihat oleh proses dan relatif terhadap ruang alamat program.

1. ***Physical address***

Physical address merupakan suatu alamat yang aktual dalam memori utama dimana data disimpan. Ini merupakan suatu lokasi dalam memori fisik, berbeda dengan alamat virtual.

1. Jelaskan bagaimana ruang memori setiap proses harus dipisahkan sebagai bentuk proteksi agar sebuah proses tidak dapat mengakses ruang memori proses yang lain. Berikan penjelasan Anda melalui gambar di bawah.



**Jawab:**

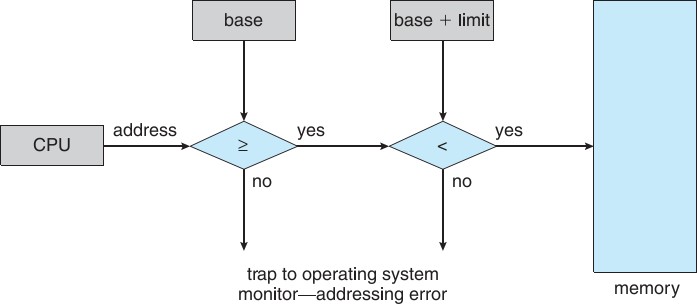
1. **Operating system (Sistem Operasi)**

Sistem operasi biasanya menempati bagian atas memori, seperti yang ditunjukkan dalam gambar oleh bagian yang disebut "operasi sistem", yang mencakup alamat 0 hingga 256000.

1. **Process (Proses)**

Proses ini memiliki ruang memori tersendiri yang dipisahkan dari proses yang lain. Dalam gambar ini, ditunjukkan dengan tiga bagian yang berlabel “Process”. Setiap proses memiliki alamat awal (base) dan akhir (limit) yang menentukan ruang memori yang dialokasikan untuk proses tersebut. Misal, proses pertama memiliki ruang memori dari 256000 hingga 300040, proses kedua dari 300040 hingga 420940, dan proses ketiga dari 420940 hingga 880000.

1. Jelaskan bagaimana alur proteksi untuk mengakses memori melalui gambar di bawah.



**Jawab:**

1. CPU menghasilkan Alamat

Proses dimulai dengan CPU yang menghasilkan alamat yang ingin diakses.

1. Pengecekan Alamat

Alamat yang dihasilkan CPU kemudian diperiksa. Ada dua kondisi yang harus dipenuhi:

* Alamat harus lebih besar atau sama dengan nilai “base”.
* Alamat harus kurang dari “base + limit”.

1. Pemberian Akses Memori

Jika kedua kondisi diatas terpenuhi, maka akses ke memori diberikan. Ini berarti bahwa proses dappat mengakses alamat memori yang diinginkan.

1. Penanganan Kesalahan

Jika salah satu atau kedua kondisi diatas tidak terpenuhi, maka terjadi kesalahan dalam penanganan alamat. Dalam hal ini, sistem operasi akan menerima perangkat dan mengindikasikan bahwa terjadi kesalahan dalam penanganan alamat.

1. Mengapa *relocation* pada manajemen memori perlu ditangani?

**Jawab:**

Karena lokasi memori tempat program ditempatkan setelah diambil kembali mungkin akan berbeda dari lokasi sebelumnya sehingga terjadi *Relocation*. Jika suatu program dimuat ke memori, alamat lokasi memori (alamat absolut atau alamat fisik) yang akan ditempati harus ditentukan. Alamar absolut pada suatu program dapat berubah-ubah sebagai akibat dari *swapping* atau *compaction*.

1. Mengapa *relocation* penting dalam manajemen memori?

**Jawab:**

Karena *Relocation* memfasilitasi alokasi yang dinamis, swapping, compaction, penggunaan memori yang optimal, proteksi, dan peningkatann kinerja sistem.

1. Mengapa *protection* perlu ditangani?

**Jawab:**

Karena untuk melindungi data dan informasi penting, mencegah akses yang tidak sah, menjaga kerahasiaan, integritas, dan ketersediaan data serta mencegah ancaman seperti virus, malware, dan serangan siber.

1. Mengapa *sharing* perlu ditangani?

**Jawab:**

Karna dapat meningkatkan efisiensi pada penggunaan memori, memfasilitasi kolaborasi dan interaksi antarindividu, serta mengelola operasi file pada sistem operasi.

1. Mengapa *local organization* dan *physical organization* perlu ditangani?

**Jawab:**

Karena untuk memastikan efisiensi dan keamanan mengakses data, serta efisiensi transfer data antara memori utama dan memori sekunder.

1. Berikan perbandingan antara *fixed partitioning*, *dynamic partitioning*, *simple paging* dan *simple segmentation*! Buatkan dalam bentuk tabel.

**Jawab:**

**Perbandingan antara *Fixed Partitioning, Dynamic Partitioning, Simple Paging* dan *Simple Segmentation.***

|  |  |  |
| --- | --- | --- |
| **Teknik** | **Kelebihan** | **Kekurangan** |
| *Fixed Partitioning* | Sederhana dan mudah diterapkan. | Dapat menghasilkan internal fragmentation dan penggunaan memori yang kurang efisien. |
| *Dynamic Partitioning* | Lebih fleksibel dan efisien dibanding *Fixed Partitioning.*. | Dapat menghasilkan external fragmentation. |
| *Simple Paging* | Menghindari fragmentation dan memudahkan manajemen memori. | Memerlukan mekanisme yang kompleks untuk mengelola table halaman. |
| *Simple Segmentation* | Memungkinkan alokasi memori yang lebih sesuai dengan kebutuhan. | Dapat menghasilkan external fragmentation. |

1. Pada *fixed partitioning* terdapat dua teknik yaitu *equal*-*size* dan *unequal*-*size*. Berikan penjelasan Anda terhadap kedua teknik tersebut dan lengkapi dengan contoh.

**Jawab:**

**Berikut penjelaasan Teknik *equal-size*  dan *unequal-size* pada *Fixed Partitioning***

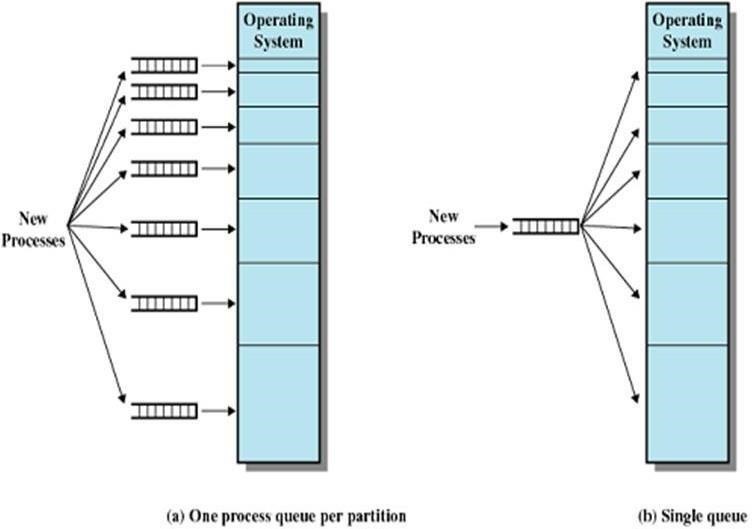
1. Equal-Size

Teknik ini membagi memori menjadi partisi dengan ukuran yang sama. Misalnya, jika memori utama berukuran 30MB, dapat dibagi menjadi lima partisi dengan ukuran masing-masing 6MB. Keuntungan dari metode ini adalah mudah digunakan dan mudah digunakan. Namun, kelemahan utamanya adalah kemungkinan fragmentasi internal, yang terjadi ketika ukuran proses lebih kecil dari ukuran partisi, sehingga ada memori yang tidak digunakan.

1. Unequal-Size

Teknik ini membagi memori menjadi partisi dengan berbagai ukuran 13. Misalnya, jika memori utama berukuran 30MB, dapat dibagi menjadi lima partisi yang masing-masing berukuran 10MB, 6MB, 4MB, 4MB, dan 6MB. Kelebihan metode ini adalah kemampuan untuk mengurangi fragmentasi internal karena ukuran partisi dapat disesuaikan dengan ukuran proses. Kerugian utamanya, bagaimanapun, adalah bahwa dapat lebih sulit untuk diurus dibandingkan dengan pembagian ukuran yang sama.

1. Pada *fixed partitioning* dengan model *unequal*-*size* terdapat dua jenis antrian seperti pada gambar di bawah. Jelaskan masing-masing model tersebut.



**Jawab:**

1. **One Process Queue per Partition**

Pada model ini, setiap partisi memiliki antrian prosesnya sendiri, yang berarti bahwa proses ditugaskan ke partisi tertentu dan menunggu giliran untuk dieksekusi di partisi yang ditugaskan. Ini memungkinkan model ini untuk mengoptimalkan penggunaan memori karena proses yang lebih kecil tidak akan mengisi partisi yang lebih besar dan meninggalkan ruang kosong.

1. **Single Queue**

Pada model ini, hanya ada satu antrian untuk semua proses baru untuk semua partisi. Proses menunggu dalam antrian umum ini dan ditugaskan ke partisi yang tersedia saat giliran mereka untuk dieksekusi. Model ini lebih sederhana dan mudah dikelola dibandingkan dengan model antrian per partisi, tetapi dapat menghasilkan internal fragmentation jika proses yang lebih kecil mengisi partisi yang lebih besar.

1. Jelaskan kelebihan dan kekurangan dari *fixed partitioning*.

**Jawab:**

**Kelebihan**

1. Sederhana dan Mudah Diterapkan

*Fixed Partitioning* ialah Teknik yang sederhana dan mudah diterapkan.

1. Dapat Mencegah Proses Mengganggu Ruang Memori Proses Lain

*Fixed Partitioning* dapat mencegah proses dari mengganggu ruang memori proses lain, sehingga meningkatkan keamanan dan stabilitas sistem.

1. Meningkatkan Keamanan Data

Dengan mempartisi hard disk, data dapat dipisahkan antara sistem operasi, data pribadi, dan file sistem. Hal ini dapat meminimalisir kerusakan pada data dan membuat data lebih aman dan terlindungi.

**Kekurangan**

1. Dapat menghasilkan Internal Fragmentation

*Fixed Partitioning* dapat menghasilkan internal fragmentation, yaitu ketika ukuran proses lebih kecil dari ukuran partisi, sehingga ada memori yang tidak terpakai.

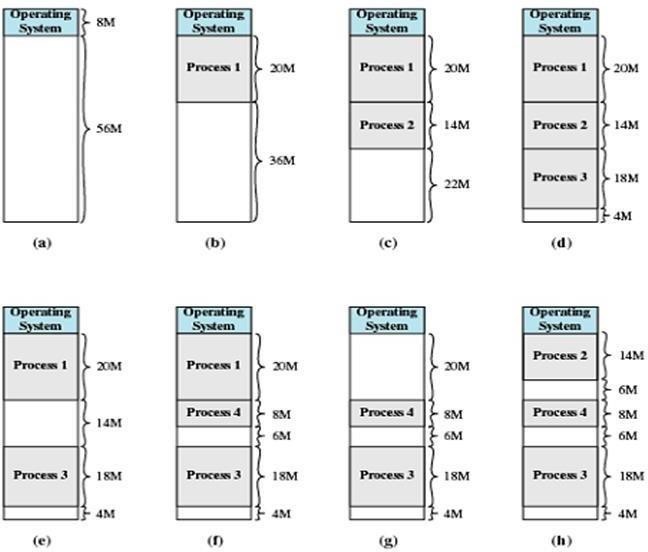
1. Membatasi Jumlah Proses yang Dapat Berjalan Secara Bersama

*Fixed Partitioning* membatasi jumlah proses yang dapat berjalan secara bersamaan, karna setiap proses memerlukan partisi yang didedikasikan.

1. Kehilangan Beberapa Ruang Hardisk untuk Area Partisi

Salah satu kerugian dari partisi Harddisk adalah kehilangan beberapa ruang Harddisk untuk area Partisi.

1. Jelaskan pengertian Anda mengenai *dynamic partitioning* melalui gambar di bawah.



**Jawab:**

Dynamic Partitioning adalah Teknik manajemen memori dimana memori utama dibagi menjadi partisi yang ukurannya dapat berubah-ubah sesuai kebutuhan.

Berikut penjelasan Dynamic Partitioning:

1. Alokasi Memori

Pada awalnya, seluruh memori (kecuali bagian yang digunakan oleh sistem operasi) tersedia untuk digunakan oleh proses-proses. Misal, pada gambar (a), ada 56M Mermori yang tersedia.

1. Penempatan Proses

Ketika proses membutuhkan memori, sistem operasi akan mencari blok memori kosong yang cukup besar untuk menampung proses tersebut. Misal, pada gambar (b) dan (c), proses 1 dan proses 2 ditempatkan di memori.

1. Pembebasan Memori

Ketika proses selesai, memori yang digunakan oleh proses tersebut akan dibebaskan dan dapat digunakan oleh proses lain. Misal, pada gambar (e), proses 1 telah selesaii dan memori yang digunakan oleh proses 1 kini tersedia untuk digunakan oleh proses lain.

1. Fragmentasi Eksternal

Selama waktu berjalan, pembebasan dan penempatan proses dapat menghasilkan blok memori kosong yang tersebar di berbagai tempat, yang dikenal sebagai fragmentasi eksternal. Misal, pada gambar (h), ada dua blok memori kosong yang terpisah.

1. Jelaskan mengapa fragmentasi eksternal dapat terjadi pada *dynamic partitioning* dan jelaskan solusinya.

**Jawab:**

Fragmentasi eksternal dapat terjadi pada dynamic partitioning arna cara alokasi memori yang dilakukan. Pada Dynamic Partitioning, memori dialokasikan berdasarkan kebutuhan spesifik setiap proses. Setelah beberapa waktu, proses-proses tersebut akan selesai dan membebaskan memori yang mereka gunakan. Namun, ruang kosong yang dihasilakn oleh proses-proses tersebut bisa saja tidak berdekatan satu sama lain. Akibatnya meskipun ada cukup ruang kosong secara total, mungkin tidak ada blok memori kosong yang cukup besar untuk menampung proses baru.

Berikut beberapa Solusi untuk mengatasi Fragmental Eksternal:

1. **Pemadatan (Compaction)**

Pemadatan (Compaction) merupakan proses memindahkan konten memori sehingga ruang kosong (fragmen) dapat digabungkan menjadi satu blok memori kosong yang berdekatan. Namun, pemadatan bisa menjadi operasi yang mahal karena memerlukan pemindahan data yang ada.

1. **Segmentasi atau Paging**

Teknik ini memungkinkan proses untuk mendapatkan memori fisik yang tidak berdekatan. Dengan cara ini, proses dapat menggunakan ruang kosong yang terpisah-pisah seolah-olah mereka berdekatan.

1. Jelaskan tiga algoritma penempatan (*placement*) yang digunakan pada *dynamic partitioning* dan urutkan ketiga algoritma tersebut dimulai dari kinerja yang paling baik hingga paling buruk.

**Jawab:**

**Berikut tiga algoritma penempatan (*placement*) yang digunakan pada *dynamic partitioning*:**

1. **First Fit**

Algoritma ini mencari blok memori kosong dari awal dan memilih blok memori yang pertama kali ditemukan dan ukurannya sesuai.

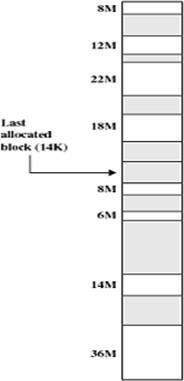
1. **Next Fit**

Algoritma ini kurang efisien dibandingkan First Fit karna blok memori yang sering ditemukan berada pada ujung akhir memori yang merupakan blok memori dengan ukuran paling besar.

1. **Best Fit**

Algoritma ini biasanya memiliki performa terburuk karena proses pencariannya paling lama dan membenani prosesor.

1. Diketahui sebuah proses baru dengan ukuran 14 MB akan diletakkan ke memori dengan gambar blok memori seperti gambar di bawah. Dengan mengikuti ketiga algoritma penempatan pada No. 17, gambarkan blok memori yang baru, beri tanda blok memori dari setiap algoritma dan sisa blok memori dari setiap algoritma.



**Jawab:**

**Berikut tanda blok memori dari setiap algoritma:**

1. **First Fit**

Algoritma ini akan mencari blok memori kosong pertama yang cukup besar untuk menampung proses baru. Dalam hal ini, blok memori kosong pertama yang cukup besar adalah blok 22M. Oleh karena itu, proses baru akan ditempatkan di blok ini, dan akan tersisa 8M di blok ini setelah penempatan.

1. **Next Fit**

Algoritma ini akan mencari blok memori kosong berikutnya setelah blok terakhir yang dialokasikan. Dalam hal ini, blok memori kosong berikutnya setelah blok terakhir yang dialokasikan (18M) adalah blok 8M, yang tidak cukup besar untuk menampung proses baru. Oleh karena itu, algoritma ini akan melanjutkan pencarian dan menemukan blok 22M, yang cukup besar untuk proses baru.

1. **Best Fit**

Algoritma ini akan mencari blok memori kosong yang paling sedikit menyisakan ruang setelah penempatan. Dalam hal ini, blok memori kosong yang paling sedikit menyisakan ruang setelah penempatan adalah blok 14M. Oleh karena itu, proses baru akan ditempatkan di blok ini, dan tidak akan ada sisa di blok ini setelah penempatan.

1. Jelaskan kelebihan dan kekurangan dari *dynamic partitioning*.

**Jawab:**

**Kelebihan**

1. **Fleksibilitas**

*Dynamic Partitioning* memungkinkan ukuran partisi untuk disesuaikan dengan kebutuhan spesifik setiap proses. Ini berarti *Dynamic Partitioning* dapat menggunakan memori secara lebih efisien dibandingkan dengan fixed partitioning.

1. **Penggunaan Memori yang Efisien**

Dengan *Dynamic Partitioning*, proses dapat menggunakan hanya sebanyak memori yang mereka butuhkan, dan memori yang tidak digunakan dapat digunakan oleh proses lain.

1. **Dapat Mengubah Volume Sistem Partisi Hard Disk Sesuai Kebutuhan**

Teknik ini dapat menambahkan partisi sebanyak yang diperlukan tanpa harus me-reboot.

**Kekurangan**

1. **Fragmentasi Eksternal**

*Dynamic Partitioning* dapat menghasilkan fragmentasi eksternal, yaitu ketika ada blok memori kosong yang tersebar di berbagai tempat dan tidak dpaat digunakan secara efisien.

1. **Kompleksitas**

*Dynamic Partitioning* dapat lebih kompleks untuk dikelola dibandingkan dengan Fixed Partitioning.

1. **Tidak Dapat Menghapus Partisipasi Hard Disk**

Untuk *Dynamic Disk,* Teknik ini tidak dapat menghapus partisipasi hard disk, jadi ini tidak dapat menghapus sistem yang dicadangkan untuk secara otomatis mengalokasikan ruang disk kosong untuk sistem.

**Praktikum**

**A. Manajemen Memori pada LINUX**

Perintah free digunakan untuk menampilkan jumlah memori fisik dan memori swap (VM), total yang belum dan yang telah digunakan oleh komputer, termasuk juga jumlah memori yang digunakan secara bersama-sama dan buffer yang digunakan oleh kernel.

Terdapat beberapa parameter yang dapat digunakan oleh perintah free. Berikut adalah parameter yang dapat digunakan antara lain:

su

-b menampilkan jumlah memori dalam byte (B)

-k menampilkan jumlah memori dalam kilobyte (KB), ini digunakan sebagai default.

-m menampilkan jumlah memori dalam megabyte (MB).

-t menampilkan sebuah baris yang berisi total.

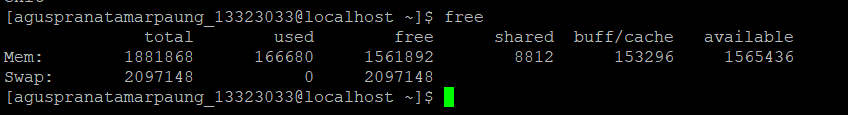
-V menampilkan informasi mengenai versi dari perintah free yang kita gunakan.

Ketikkan perintah free dan output eksekusi perintah tersebut. Lakukan langkah-langkah berikut dan amati hasilnya:

1. Ketikkan perintah **free**

|  |
| --- |
| sanhenra@ubuntu:~$ free total used free shared buffers cached  Mem: 1016832 320204 696628 0 13716 148336  -/+ buffers/cache: 158152 858680  Swap: 1646620 0 1646620 |

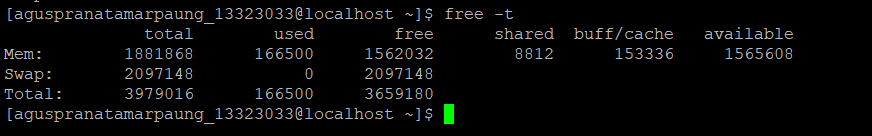
**Hasil:**

****

1. Ketikkan perintah **free** dengan menambahkan opsi **-t**

|  |
| --- |
| sanhenra@ubuntu:~$ free -t total used free shared buffers cached  Mem: 1016832 338780 678052 0 14008 163228  -/+ buffers/cache: 161544 855288  Swap: 1646620 0 1646620  Total: 2663452 338780 2324672 |

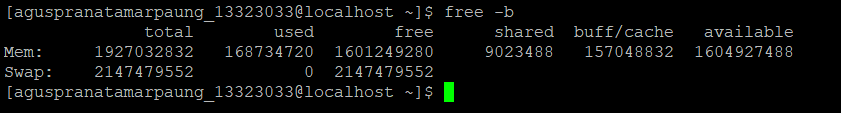
**Hasil:**

****

1. Ketikkan perintah **free** dengan menambahkan opsi **-b**

|  |  |
| --- | --- |
| sanhenra@ubuntu:~$ free -b total used free shared buffers cached  Mem: 1041235968 346853376 694382592  -/+ buffers/cache: 165355520 875880448  Swap: 1686138880 0 1686138880 | 0 14352384 167145472 |

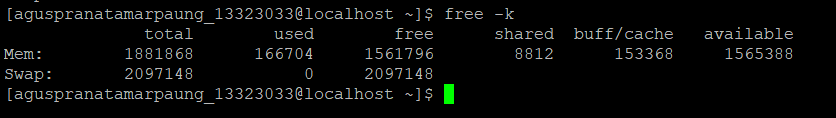
**Hasil:**

****

1. Ketikkan perintah **free** dengan menambahkan opsi **-k**

|  |
| --- |
| sanhenra@ubuntu:~$ free -k total used free shared buffers cached  Mem: 1016832 338724 678108 0 14016 163228  -/+ buffers/cache: 161480 855352  Swap: 1646620 0 1646620 |

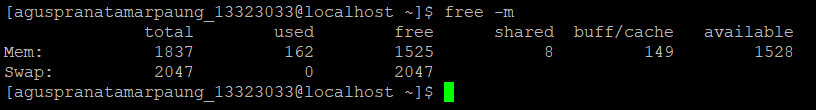
**Hasil:**

****

1. Ketikkan perintah **free** dengan menambahkan opsi **-m**

|  |
| --- |
| sanhenra@ubuntu:~$ free -m  total used free shared buffers cached Mem:  993330 662 0 13 159  -/+ buffers/cache: 157 835  Swap: 1608 0 1608 |

**Hasil:**

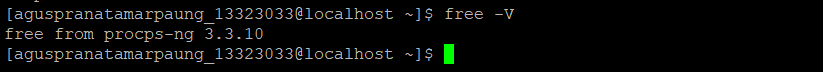
****

1. Ketikkan perintah **free** dengan menambahkan opsi **-V**

sanhenra@ubuntu:~$ free -V

procps version 3.2.7

**Hasil:**

****

Sekarang anda bandingkan hasil yang anda peroleh tadi dengan isi file /proc/meminfo. File /proc/meminfo ini berisi tentang informasi memori yang terdapat pada komputer kita.

1. Amati isi file /proc/meminfo, apa kesimpulan Anda?

[root@si ~]# cat /proc/meminfo

MemTotal: 2059440 kB

MemFree: 293156 kB

Buffers: 178696 kB

Cached: 1042952 kB

SwapCached: 0 kB

Active: 1045888 kB Inactive: 506152 kB

HighTotal: 0 kB

HighFree: 0 kB

LowTotal: 2059440 kB

LowFree: 293156 kB

SwapTotal: 2048248 kB

SwapFree: 2048144 kB

Dirty: 196 kB

Writeback: 0 kB

AnonPages: 330372 kB

Mapped: 59128 kB

Slab: 163644 kB

PageTables: 28256 kB

NFS\_Unstable: 0 kB

Bounce: 0 kB

CommitLimit: 3077968 kB

Committed\_AS: 1390340 kB

VmallocTotal: 34359738367 kB

VmallocUsed: 263888 kB

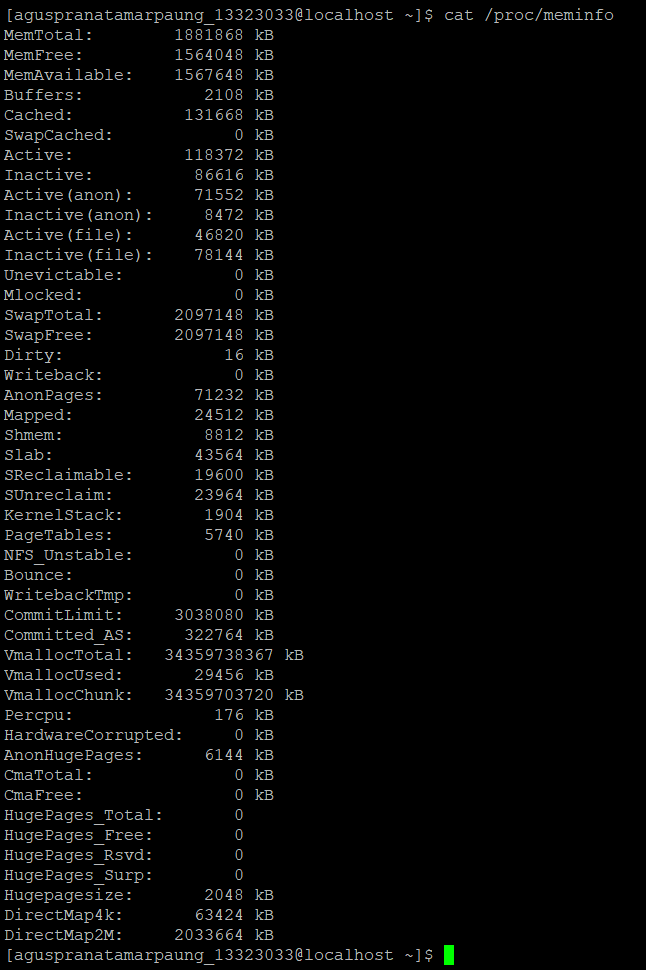
VmallocChunk: 34359473775 kB

HugePages\_Total: 0 HugePages\_Free: 0

HugePages\_Rsvd: 0

Hugepagesize: 2048 kB

**Hasil:**

****

**Kesimpulan:**

Dari output diatas, total memori fisik yang tersedia dalam sistem anda adalah sekitar 1,8 GB (*MemTotal*), dengan sekitar 1,5 GB yang tidak digunakan (*MemFree*). Memori yang tersedia untuk digunakan oleh aplikasi tanpa mempengaruhi kinerja sistem juga sekitar 1,5 GB (*MemAvailable*). Memori yang digunakan oleh kernel untuk buffer adalah 21608 kB (*Buffers*) dan memori yang digunakan oleh page cache dan slabs adalah 131668 kB (*Cached*). Tidak ada halaman yang ditukar, yang berarti sistem ini tidak mengalami tekanan memori. Total ruang swap yang tersedia adalah sekitar 2 GB (*SwapTotal*), dan semua ruang swap ini bebas digunakan.

**B. Manajemen memori pada DOS**

Untuk memeriksa memori yang terdapat pada komputer yang menggunakan sistem operasi DOS (Disk Operating Systems) dapat menggunakan perintah MEM. Perintah MEM akan menampilkan informasi tentang daerah memori yang telah dialokasikan, daerah memori yang belum digunakan, dan program-progam yang telah dimuat. Selain itu, informasi lain dapati diperoleh dengan cara menyertakan parameter tambahan pada saat memberikan perintah.

**PERHATIAN:**

Parameter-parameter yang disajikan di sini dapat berjalan di DOS 98 atau sebelumnya, seperti DOS 6.0. Apabila anda menjalankan DOS emulator pada sistem operasi Windows XP maka tidak semua parameter di bawah ini berjalan dengan baik. Anda dapat mengetahui parameter apa saja yang tersedia beserta keterangan fungsinya dengan menjalankan perintah berikut pada shell prompt anda:

1. Ketikkan perintah **mem** dengan menambahkan opsi **/?**

C:\Users\good>mem /?

Displays the amount of used and free memory in your system.

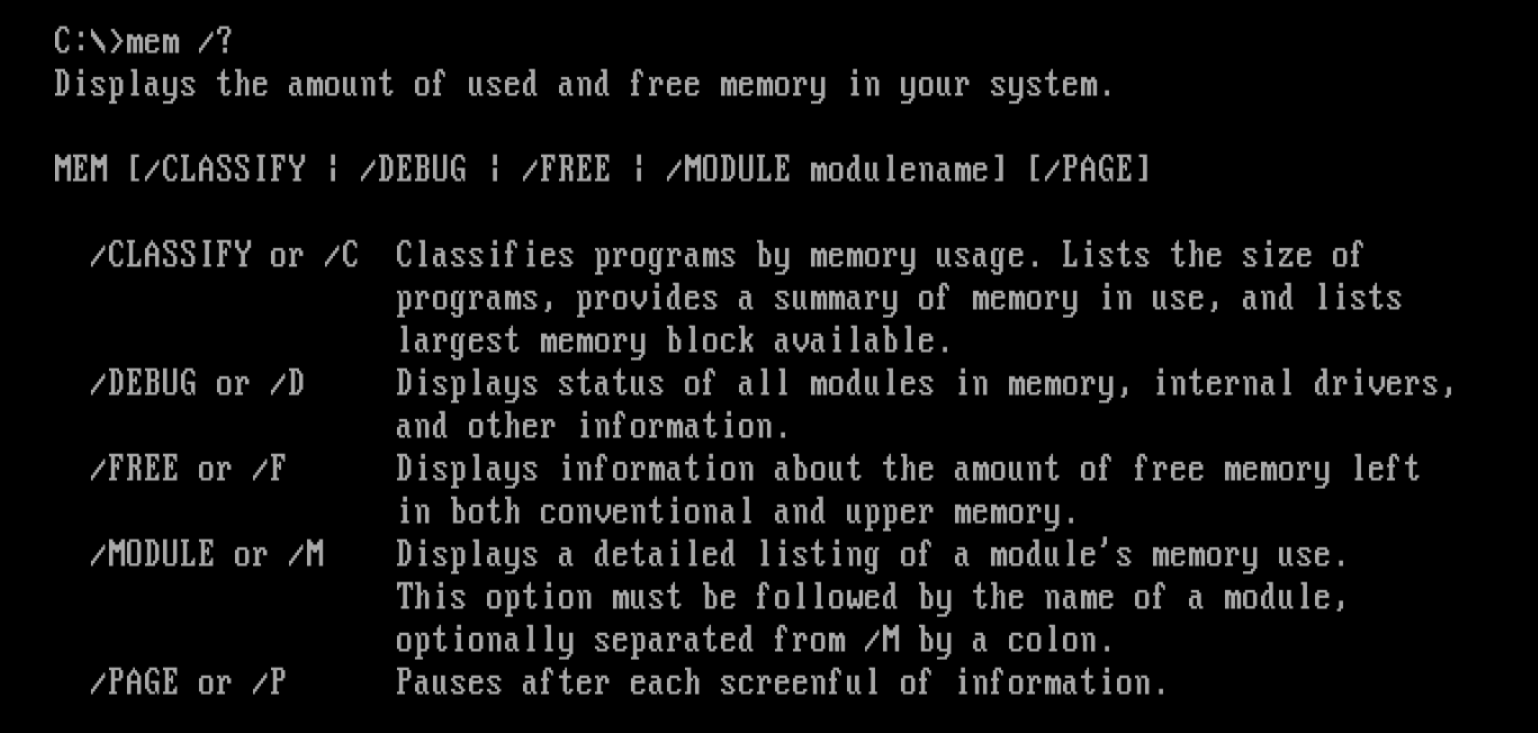
MEM [/PROGRAM | /DEBUG | /CLASSIFY]

/PROGRAM or /P Displays status of programs currently loaded in memory.

/DEBUG or /D Displays status of programs, internal drivers, and other information.

/CLASSIFY or /C Classifies programs by memory usage. Lists the size of programs, provides a summary of memory in use, and lists largest memory block available.

**Hasil:**

****

**Penjelasan:**

Fungsi sintaks mem /? yaitu sebagai berikut:

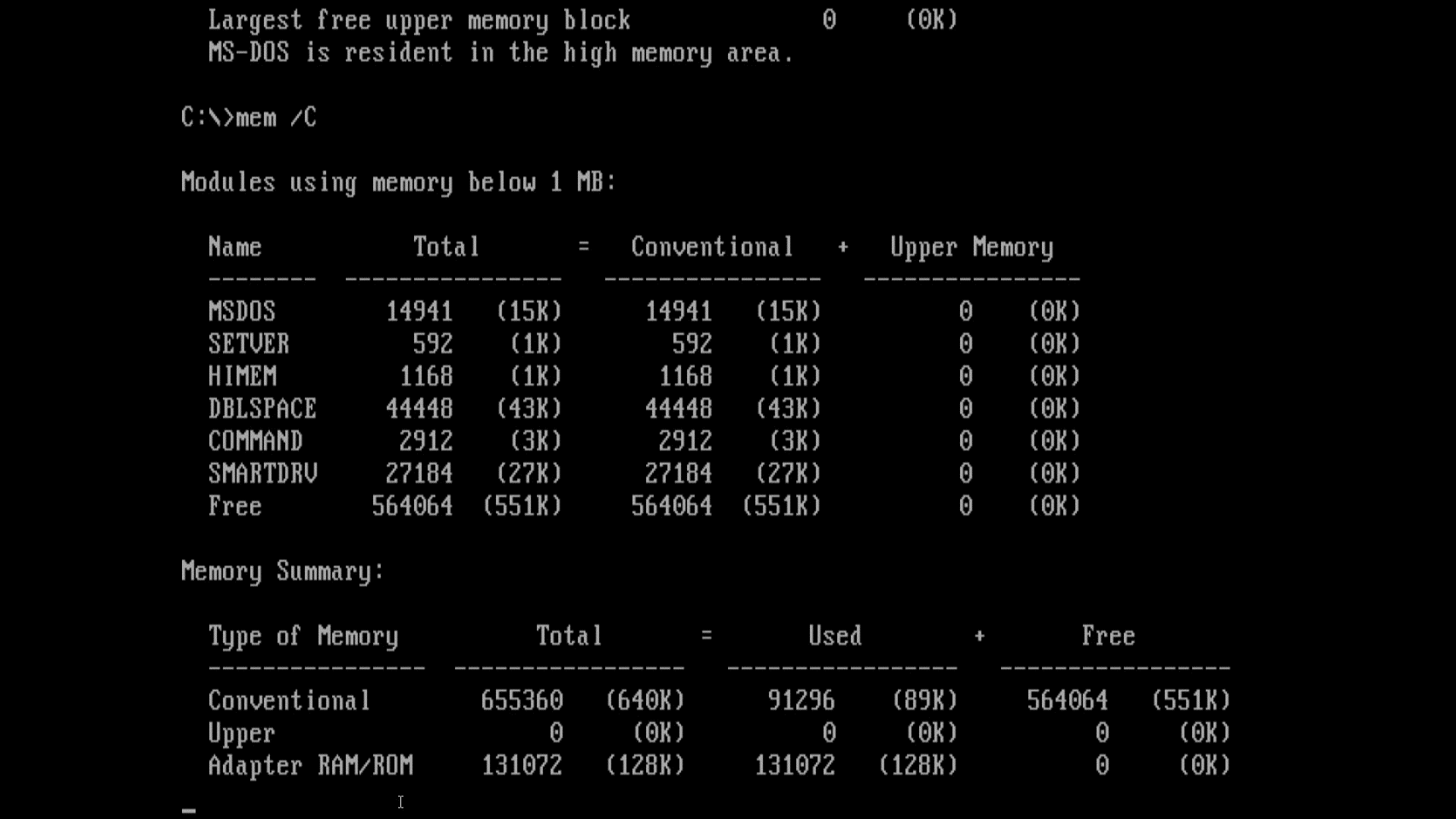
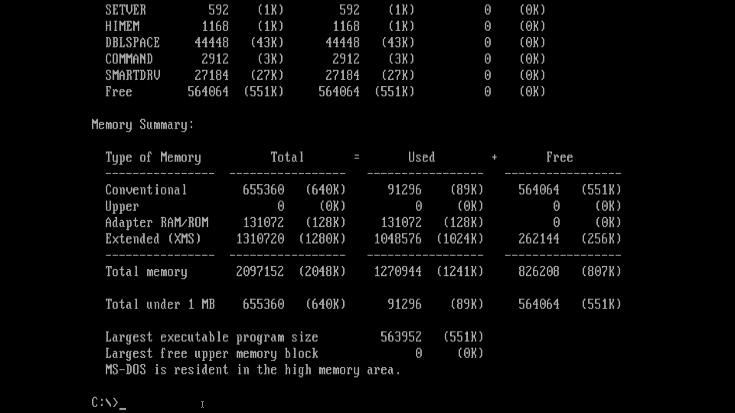
**mem =**  untuk menampilkan informasi yang merujuk ke memori.

**/?** **=** sebagai bantuan atau petunjuk ketika tidak yakin tentang bagaimana menggunakan perintah tertentu.

1. Ketikkan perintah **mem** dengan menambahkan opsi **/CLASSIFY atau /C.** Menginformasikan program-program yang telah dimuat ke dalam memori dan jumlah memori yang digunakan masing-masing program tersebut. Selain itu juga ditampilkan informasi seperti yang ditampilkan pada perintah MEM tanpa parameter.

|  |  |
| --- | --- |
| C:\Users\good>mem /C    Conventional Memory :  Name Size in Decimal Size in Hex  MSDOS 12288 ( 12.0K) 3000  KBD 3360 ( 3.3K) D20  HIMEM 1248 ( 1.2K) 4E0  COMMAND 3968 ( 3.9K) F80  DOSX 34704 ( 33.9K) 8790  FREE 112 ( 0.1K) 70  FREE 599488 (585.4K) 925C0  Total FREE : 599600 (585.5K)    Upper Memory :  Name Size in Decimal Size in Hex  SYSTEM 200688 (196.0K) 30FF0  DOSX 128 ( 0.1K) 80  MOUSE 12528 ( 12.2K) 30F0  MSCDEXN 352 ( 0.3K) 160  T REDIR 2176 ( 2.1K) 880  FREE 1168 ( 1.1K) 490  FREE 44976 ( 43.9K) AFB0  Total FREE : 46144 ( 45.1K)  Total bytes available to programs (Conventional+Upper) : 645744 (630.6K)  Largest executable program size : 598288 (584.3K) | |
| Largest available upper memory block :  1048576 bytes total contiguous extended memory 0 bytes available contiguous extended memory  941056 bytes available XMS memory  MS-DOS resident in High Memory Area | 44976 ( 43.9K) |

**Hasil:**



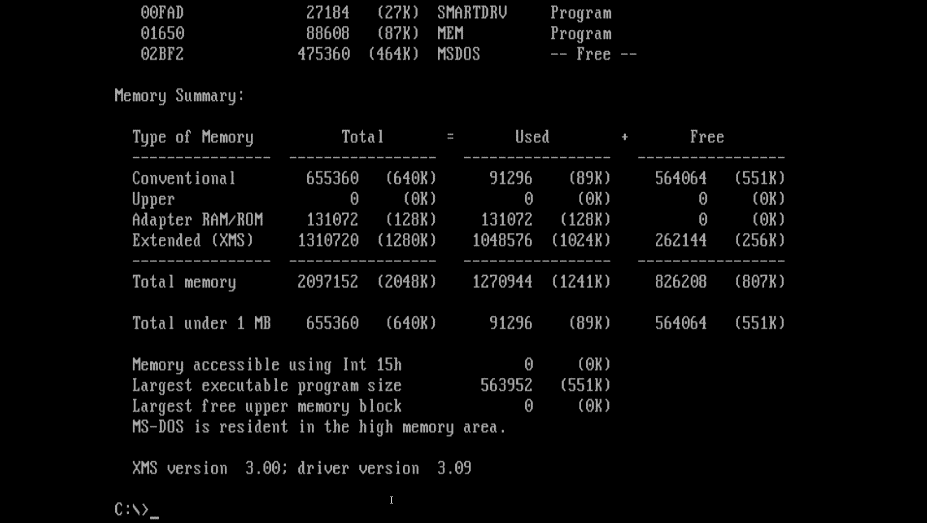
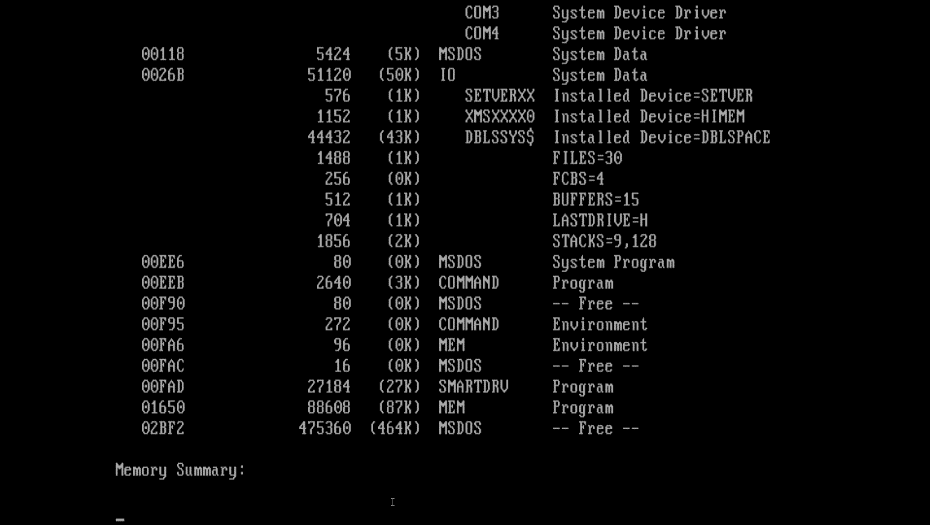
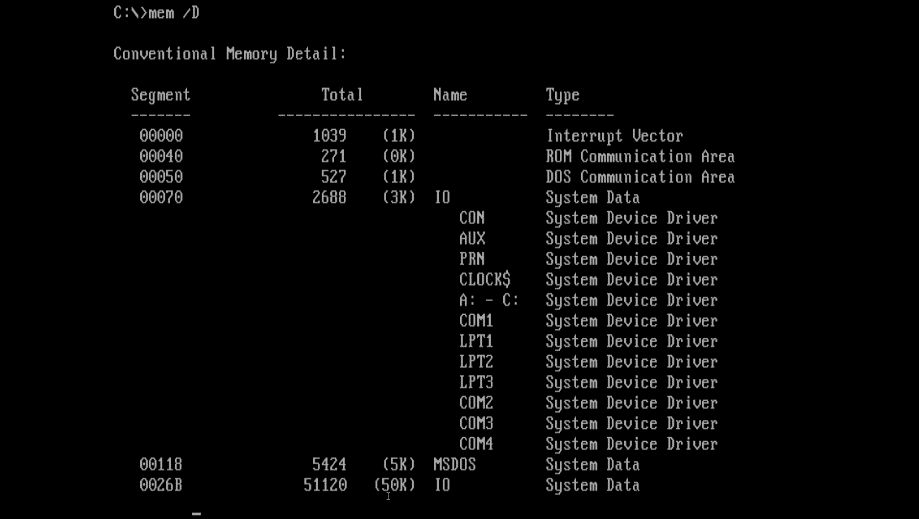
**Penjelasan:**

Fungsi mem /C yaitu unutk menampilkan rasio pengkompresan dan memberikan informasi tentang sejauh mana data telah dikompresi, yang berguna untuk mendiagnosis masalah performa atau memahami penggunaan ruang disk.

1. Ketikkan perintah **mem** dengan menambahkan opsi **/DEBUG atau /D**. Menampilkan daftar program dan driver yang telah dimuat ke dalam memori. Juga ditampilkan ukuran setiap modul, alamat segmen, jenis modul, ringkasan Ketikkan memori secara keseluruhan, dan informasi lain yang sangat berguna untuk pemrograman.

|  |
| --- |
| C:\Users\good>mem /D    Address Name Size Type  000000 000400 Interrupt Vector  000400 000100 ROM Communication Area  000500 000200 DOS Communication Area  000700 IO 000370 System Data CON  System Device Driver  AUX System Device Driver  PRN System Device Driver  CLOCK$ System Device Driver  COM1 System Device Driver  LPT1 System Device Driver  LPT2 System Device Driver  LPT3 System Device Driver  COM2 System Device Driver  COM3 System Device Driver  COM4 System Device Driver  000A70 MSDOS 001690 System Data    002100IO 002150 System Data KBD  000D20 System Program HIMEM  0004E0 DEVICE=  XMSXXXX0 Installed Device Driver  000490 FILES=  000090 FCBS=  000200 LASTDRIVE=  0007D0 STACKS=  004260 COMMAND 000A20 Program  004C90 MSDOS 000070 -- Free -- 004D10 COMMAND 000560 Environment  005280 DOSX 008790 Program  00DA20 MEM 0004A0 Environment  00DED0 MEM 0174E0 Program  0253C0 MSDOS 07AC20 -- Free -- 09FFF0 SYSTEM 031000 System Program    0D1000 IO 003100 System Data MOUSE 0030F0 System Program  0D4110 MSDOS 000490 -- Free -- 0D45B0 MSCDEXNT 000160 Program  0D4720 REDIR 000880 Program  0D4FB0 DOSX 000080 Data  0D5040 MSDOS 00AFB0 -- Free --      655360 bytes total conventional memory 655360 bytes available to MS-DOS 598288 largest executable program size    1048576 bytes total contiguous extended memory 0 bytes available contiguous extended memory  941056 bytes available XMS memory  MS-DOS resident in High Memory Area |

**Hasil:**

****

**Penjelasan:**

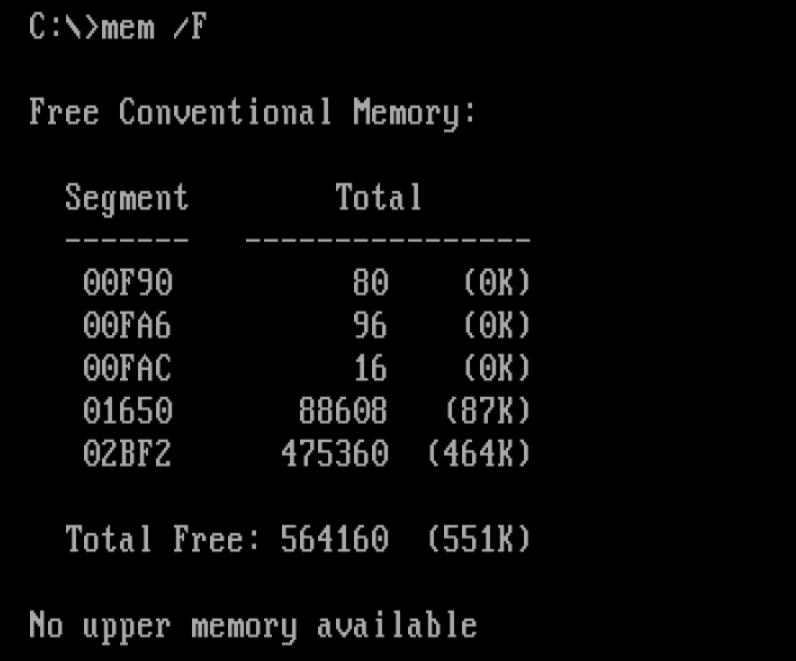
Fungsi mem /D yaitu untuk menampilkan daftar yang sedang berjalan, termasuk alamat memori, ukuran, dan jumlah memori yang digunakan dan juga berguna untuk debugging atau optimasi dan melihat secara langsung bagaimana memori sistem digunakan.

1. Ketikkan perintah **mem** dengan menambahkan opsi **/FREE atau /F.**

Menampilkan daerah yang masih belum digunakan pada memori konvensional dan upper memory area. Juga ditampilkan alamat segment dan ukuran masing-masing daerah yang masih kosong tersebut.

C:\Users\good>mem /F

**Hasil:**

****

**Penjelasan:**

Fungsi mem /D yaitu untuk menampilkan daftar semua blok memori, termasuk blok yang kosong (tidak digunakan), dan menunjukkan ukuran dan lokasi setiap blok dan juga berguna untuk mendiagnosis masalah memori dan melakukan optimasi dan melihat secara langsung bagaimana memori sistem digunakan.

Cobalah opsi lainnya dan jelaskan kegunaan masing-masing opsi tersebut!