Memory Management

Memory Management Requirement
- relocation
- protection, logical & physical address space
- sharing
- logical organization
- physical organization

Relocation
- process of assigning load addresses to the various parts of prgm, adjusting code & data in prgm to reflect the assigned addresses
- relocation modifies object prgm so that it can be loaded at an address ≠ from location originally specified

Logical & physical address space
- concept of logical address space that is bound to a separate physical address space is central to proper memory mgment
  - logical address
    - generated by CPU
    - referred to as virtual address
  - physical address
    - address seen by memory unit
- logical & physical address same at compile time & load time address binding scheme
- logical (virtual) and physical addresses differ in execution time address binding scheme

Protection
- Keeps the process from accessing the address space of another process

Sharing
- protection mechanism should allow many processes to access same part of memory
- all the processes executing same prgm should be allowed to use single copy of prgm
- co-ordinating process should be allowed to use same data structure and controlled access to it should be ensured

Logical Organization
processes are composed of modules of varying sizes, modules independently compiled, modules with ≠ protection needs even to the degree of sharing

Physical Organization
- 2 levels of organization available in system
  - main memory
  - secondary memory
- Main Memory (RAM) offers fast access than 2ndary memory with high cost & volatile
- Secondary memory provides permanent storage
- Info exchanged between main memory and secondary memory should be handled by system and not the programmer

Memory Partition
- monoprogramming
- multiprogramming

Monoprogramming

User program
OS

- In early computer system & early micro computer OS monoprogramming was used
- Only one prgm was residing in main memory at given pt of time
- OS was residing in some portion of memory & other other portion was fully devoted the single executing process
- simple design
- not supported multiprogramming

Multiprogramming
- required to support multiple process simultaneously
- since multiple processes are resident in the memory at same time, it increases processor utilization if processes are I/O bound, CPU doesn't stay idle
- CPU utilization increases as it executes multiple processes on time sharing basis

Memory Partitioning - Fixed Partitioning (Contiguous Allocation)
- memory is divided into a no. of fixed size partition, each partition may contain exactly one process

Difficulties
- Programmer may have to re-write a prgm in order to fit into even the largest available partition, overlaying may be used
- Internal fragmentation typically results - Memory that is unused within each partition

Internal & External Fragmentation

Internal Fragmentation	External Fragmentation
- takes place in fixed partitioning technique	- takes place in variation partitioning technique
- difference memory allocated & required memory ⇒ internal fragmentation	- unused space formed between non contiguous memory fragments are too small to serve a new process request ⇒ external fragmentation
- occurs when main memory is divided into fixed size blocks regardless of size of process	- occurs when main memory is allocated to processes dynamically based on process request
- it can eliminated by allocating memory to process dynamically	- it can be eliminated by compacting, segmentation & paging

Compaction
- technique to convert many scattered holes into one large size hole
- it's relocation of various processes so as to accumulate all free space in one place
- Due to compaction, efficient use of processor

Swapping
- system in which it's possible to move entire processes between disk & main memory during execution

Memory allocation strategies
- First Fit
- Best Fit
- Worst Fit
- Next Fit

First Fit
- process is allocated in the first available memory space that is large enough for its purpose
- Searching can start at the beginning or at the pt where first fit search ended

Best Fit
- process is allocated in the smallest memory block that is large enough for its purpose
- for best fit algo, entire block list needs to be searched unless maintained in order

Worst Fit
- process is allocated in the largest memory block that is available
- for worst fit algo, entire block list needs to be searched unless maintained in order

Next Fit
- starts the searching of memory from position of previous placement & selects the next available block that is large enough

Buddy system
- compromise between fixed & dynamic partitioning
- while internal fragmentation occurs to a degree external fragmentation is eliminated & no pre determined block size is established

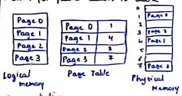
Relocation
- When a process enters the system or reenters after swapping process, it may not be placed into physical memory same as before

Virtual Memory
- a computer can address more memory than amount physically installed on system. This extra memory → Virtual Memory
- section of hardisk that setup to emulate computer's RAM
- Adv: program can be larger than physical memory

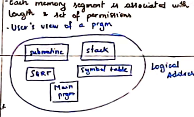
Dynamic Loading
- Ensure the better memory space utilization unless and until called, routine is not loaded in main memory
- All routines remain stored on disk in relocatable load format
- Main prgm is loaded in main memory & executed
- No special support required from OS

Overlays
- In Early days, large prgms were too large to fit in the partition, programmers created overlays to solve this pb
- Rather than loading an entire prgm into main memory at once, we can divide it in modules that are always required at certain times during prgm execution
- Overlays involve moving data & prgm segment in & out of memory which helps for reusing area in main memory
- Physical division

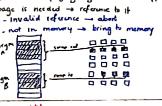
Paging
- memory management technique
- memory is divides into fixed size pages
- O.S retrieves data from secondary storage in same size blocks called pages.
- important part of virtual memory implement in modern OS
- it allows physical address space of process to be non contiguous
- concept is used to remove pb of fragmentation
- used for faster access to data



Segmentation
- It is one of the most common way to achieve memory protection
- Because of internal fragmentation of pages takes place, the user's view of memory is lost
- The user will view the memory as the combination of segment
- Memory address used are not contiguous
- Each memory segment is associated with a length & set of permissions
- User's view of a prgm



Demand Paging
- could bring entire process into memory at load time
- bring a page into memory only when its needed
  - less I/O needed
  - less memory needed
  - faster response
  - more users
- similar to paging system with swapping
- page is needed → reference to it
  - invalid reference → abort
  - not in memory → bring to memory



Copy on right
- allows parent & child to share the same pages
- if parent or child writes to shared pages then it is marked as copy on right & its copy is created by fork() system call

Page replacement strategies
- when process generate a page fault, memory manager must locate referenced page in secondary storage, load it into page frame in main memory & update corresponding page table entry

Structure of Page Table
- Hierarchical paging
- Hashed page table
- Inverted page table

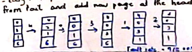
Page table
- data structure used by a virtual memory system in a computer operating system
- used to store the mapping between virtual & physical addresses

Hierarchical paging
- most modern computer system support a large logical address space (2^{32} to 2^{64})
- in such environment page table itself becomes excessively large
- to solve this pb, computer use hierarchical paging technique
- known as multilevel paging
- the page table might be too big to fit in a contiguous space, so we may have hierarchy with several levels
- in this, logical address space is broken into multiple space tables.

Hashed page table
- if address spaces are larger than 32 bit then hashed page table is used in which hashed value is used as virtual space number
- elements which are hashed at same location its linked list is maintained at each entry in hash table
- necessary to avoid collision
- each element consist of -
  - virtual page number
  - value of mapped page frame
  - a pointer to the next element in L.L

Inverted page table
- O.S converts the page reference into physical memory address
- because table is sorted by virtual address, OS capable of computing where in table the associated physical address entry is located and to use that value directly
- Disadv: each page table contains millions of entries
- these tables may use large amount of physical memory just to monitor how other physical memory is being used
- pb solved by inverted page tables

FIFO Algo (First in First out)
- oldest page in main memory is one which will be selected for replacement
- easy to implement, keep a list, replace pages from tail and add new page at the head

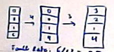


Reference String: 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1
Misses

Fault rate = 9/12 = 0.75

Optimal page algo
- an optimal page replacement algo has lowest page fault rate of all algos
- an optimal page replacement algo exist and has been called OPT or MIN
- replace the page that will not be used for longest period of time
- used the time when page is to be used

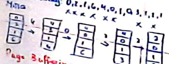
Reference String: 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1
Misses: × × × × × ×



Fault Rate: 6/12 = 0.5

LRU Algo (Least Recently Used)
- page which has not been used for the longest time in main memory is the one which will be select for replacement
- easy to implement, keep a list, replace page by looking back into time

Reference String: 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1
Misses: × × × × × × × ×



Page Buffering Algo
- to get a process start quickly keep a pool of free frames
- on page fault, select page to be replaced
- write the new page in frame of free pool, mark page table & restart process
- Now write dirty page out of disk & place frame holding replaced page in free pool

LFU Algo (Least Frequently Used)
- page with smallest count is the one which will be selected for replacement
- algo suffers from situation in which a page is used heavily during initial phase of a process but then is never used again

MFU Algo (Most Frequently Used)
- algo based on argument that page with smallest count was probably just brought in & has yet to be used

Allocation of Frames
- OS maintains free frame list.
- Single user system with 512 KB of memory consisting of pages 4 KB in size.
- System has 128 frames each of 4 KB in OS may occupy 35 frames and remaining 93 frames are used by other process
  - Equal Allocation
  - Proportional Allocation
  - Global v/s Local Allocation

Thrashing
- if a process doesn't have enough pages, the page fault rate is very high.
- this leads to
  - low CPU utilization
  - O.S thinks that it needs to increase the degree of multiprogramming
  - another process added to system

✓ A process is busy swapping pages in & out is called as Thrashing

Memory Management

Memory Management Requirment

- relocation
- protection, logical + physical address space
- sharing
- logical organization
- physical organization

Relocation

- process of assigning load addresses to the various parts of prgm, adjusting code & data in prgm to reflect the assigned addresses
- relocation modifies object prgm so that it can be loaded at an address ≠ from location originally specified

Logical & physical address space

- concept of logical address space that is bound to a separate physical address space is central to proper memory mgment
  - logical address
    - generated by CPU
    - refered to as virtual address
  - physical address
    - address seen by memory unit
- logical & physical address same at compile time & load time address binding scheme
- logical (virtual) and physical addresses differ in execution time address binding scheme

Protection

- keeps the process from accessing the address space of another process

Sharing

- protection mechanism should allow many processes to access same part of memory
- all the processes executing same prgm should be allowed to use single copy of prgm
- co-ordinating process should be allowed to use same data structure and controlled access to it should be ensured

Logical Organization

- processes are composed of modules of varying sizes, modules independently compiled, modules with ≠ protection needs even to the degree of sharing

Physical organization

- 2 levels of organization available in system
  - main memory
  - secondary memory
- Main Memory (RAM) offers fast access than 2ndary memory with high cost & ...
- Secondary memory provides permanent storage
- Info exchanged between main memory and secondary memory should be handled by system and not the programmer

Memory Partition

- monoprogramming
- multiprogramming

Monoprogramming

User program
OS

- In early computer system & early micro computer as monoprogramming was used
- Only one prgm was residing in main memory at given pt of time
- OS was residing in some portion of memory & other other portion was fully devoted the single executing process
- simple design
- not supported multiprogramming

Multiprogramming

- required to support multiple process simultaneously
- since multiple processes are resident in the memory at same time, it increases processor utilization if processes are I/O bound, CPU doesn't stay idle
- CPU utilization increases as it executes multiple processes on time sharing basis

Memory Partitioning - Fixed Partitioning (Contiguous Allocation)

- memory is divided into a no. of fixed size partition, each partition may contain exactly one process

Difficulties

- Programmer may have to re-write a prgm in order to fit into even the largest available partition, overlaying may be used
- Internal fragmentation typically results - Memory that is unused within each partition

Internal & External Fragmentation

Internal Fragmentation	External Fragmentation
- takes place in fixed partitioning technique	- takes place in variation partitioning technique
- difference memory allocated & required memory ⇒ internal fragmentation	- unused spaces formed between non contiguous memory fragments are too small to serve a new process request ⇒ external fragmentation
- occurs when main memory is divided into fixed size block regardless of size of process	- occurs when main memory is allocated to processes dynamically based on process request
- it can eliminated by allocating memory to process dynamically	- it can be eliminated by compacting, segmentation & paging

Compaction

- technique to convert many scattered holes into one large size hole
- it's relocation of various processes so as to accumulate all free space in one place
- Due to compaction, efficient use of processor

Swapping

- system in which it's possible to move entire processes between disk & main memory during execution

Memory allocation strategies

- First Fit
- Best Fit
- Worst Fit
- Next Fit

First Fit

- process is allocated in the first available memory space that is large enough for its purpose
- Searching can start at the beginning or at the pt where first fit search ended

Best Fit

- process is allocated in the smallest memory block that is large enough for its purpose
- for best fit algo, entire block list needs to be searched unless maintained in order

Worst Fit

- process is allocated in the largest memory block that is available
- for worst fit algo, entire block list needs to be searched unless maintained in order

Next Fit

- starts the searching of memory from portion of previous placement & selects the next available block that is large enough

Buddy system

- compromise between fixed & dynamic partitioning
- while internal fragmentation occurs to a degree
- external fragmentation is eliminated & no pre determined block size is established

Relocation

- When a process enters the system or reenters after swapping process, it may not be placed into physical memory same as before

Virtual Memory

- a computer can address more memory than amount physically installed on system. This extra memory → Virtual Memory
- section of hardisk that setup to emulate computer's RAM
- Adv: program can be larger than physical memory

Dynamic Loading

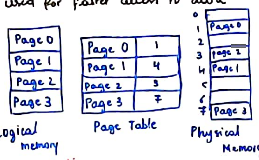
- Ensure the better memory space utilization unless and until called, routine is not loaded in main memory
- All routines remain stored on disk in relocatable load format
- Main prgm is loaded in main memory & executed
- No special support required from OS

Overlays

- In Early days, large prgms were too large to fit in the partition, programmers created overlays to solve this pb
- Rather than loading an entire prgm into main memory at once, we can divide it in modules that are always required at certain times during prgm execution
- Overlays involve moving data & prgm segment in & out of memory which helps for for reusing area in main memory
- Physical division

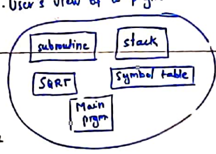
Paging

- memory management technique
- memory is divides into fixed size pages
- O.S retrieves data from secondary storage in same size blocks called pages.
- important part of virtual memory implementation in modern OS
- it allows physical address space of process to be non contiguous
- concept is used to remove pb of fragmentation
- used for faster access to data



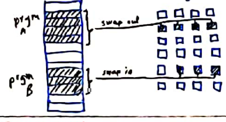
Segmentation

- It is one of the most common way to achieve memory protection
- Because of internal fragmentation of pages takes place, the user's view of memory is lost
- The user will view the memory as the combination of segment
- Memory address used are not contiguous
- Each memory segment is associated with a length & set of permissions
- User's view of a prgm



Demand Paging

- could bring entire process into memory at load time
- bring a page into memory only when its needed
  - less I/O needed
  - less memory needed
  - faster response
  - more users
- Similar to paging system with swapping
- page is needed → reference to it
  - Invalid reference → abort
  - not in memory → bring to memory



Copy on right

- allows parent & child to share the same pages
- if parent or child writes to shared pages then it's marked as copy on right & its copy is created by fork() system call

Page replacement strategies

- when process generates a page fault, memory manager must locate referenced page in secondary storage, load it into page frame in main memory & update corresponding page table entry

… of Page Table

- … al paging
- Hashed page table
- Inverted page table

Page table

- data structure used by a virtual memory system in a computer operating system
- used to store the mapping between virtual & physical addresses

Hierarchical paging

- most modern computer system support a large logical address space (2^{32} to 2^{64})
- in such environment page table itself become excessively large
- to solve this pb, computer uses hierarchical paging technique
- known as multilevel paging
- the page table might be too big to fit in a contiguous space, so we may have hierarchy with several levels
- in this, logical address space is broken into multiple space tables.

Hashed page table

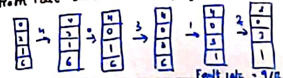
- if address spaces are larger than 32 bit then hashed page table is used in which hashed value is used as virtual space number
- elements which are hashed at same location, its linked list is maintained at each entry in hash table
- necessary to avoid collision
- each element consist of -
  - virtual page number
  - value of mapped page frame
  - a pointer to the next element in L.L

Inverted page table

- O.S converts the page reference into physical memory address
- because table is sorted by virtual address, OS capable of computing where in table the associated physical address entry is located and to use that value directly
- Disadv: each page table contains millions of entries
- these tables may use large amount of physical memory just to monitor how other physical memory is being used
- pb solved by inverted page tables

FIFO Algo (FIRST IN FIRST OUT)

- oldest page in main memory is one which will be selected for replacement
- easy to implement, keep a list, replace pages from tail and add new page at the head



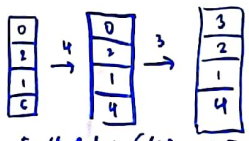
Fault rate = 9/12 = 0.75

Reference String: 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1
Misses: X X X X X X X

Optimal page algo

- an optimal page replacement algo has lowest page fault rate of all algos.
- an optimal page replacement algo exist and has been called OPT or MIN
- replace the page that will not be used for longest period of time
- used the time when page is to be used

Reference String: 0,2,1,6,4,0,1,0,3,1,2,1
Misses: x x x x x x

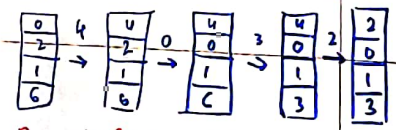


Fault Rate: 6/12 = 0.5

LRU Algo (Least Recently used)

- page which has not been used for the longest time in main memory is the one which will be select for replacement
- easy to implement, keep a list, replace page by looking back into time

Reference String: 0,2,1,6,4,0,1,0,3,1,2,1
Misses: x x x x x x x x



Page Buffering Algo

- to get a process start quickly, keep a pool of free frames
- on page fault, select a page to be replaced
- write the new page in frame of free pool, mark page table & restart process
- Now write dirty page out of disk & place frame holding replaced page in free pool

LFU Algo (Least Frequently used)

- page with smallest count is the one which will be selected for replacement
- algo suffers from situation in which a page is used heavily during initial phase of a process, but then is never used again

MFU Algo (Most Frequently used)

- algo based on argument that page with smallest count was probably just brought in & has yet to be used.

Allocation of frames

- OS maintains free frame list.
- Single user system with 512 KB of memory consisting of pages 4 KB in size.
- System has 128 frames each of 4 KB in size. OS may occupy 35 frames and remaining 93 frames are used by other process
  - Equal Allocation
  - Proportional Allocation
  - Global v/s Local Allocation

Thrashing

- if a process doesn't have enough pages, the page fault rate is very high.
- this leads to
  - low CPU utilization
  - OS thinks that it needs to increase the degree of multiprogramming
  - another process added to system

✓ A process is busy swapping pages in & out is called as thrashing