MEMORY MANAGEMENT IN UNIX

STUDENT NAME : AMAL ISWIASI

STUDENT NO: 153104020

PRESENT TO : Prof. Dr. Hasan Huseyin Balik

contents

Introduction to UNIX

- UNIX is a portable, multi-tasking and multi-user operating system.
  - Portable: runs on many different hardware architectures (Intel x86, HP PA-RISC, PowerPC, IBM S/390, etc.).
  - Preemptive multi-tasking: several programs can run at the same time (time slices, interrupts, and task switching).
  - Multi-user: many users can share the computer system at the same time.

UNIX Memory Management

- Memory is an important resource in computer.
- Memory management is the process of managing the computer memory which consists of primary memory and secondary memory.
- The goal for memory management is to keep track of which parts of memory are in use and which parts are not in use, to allocate memory to processes when they need it and de-allocate it when they are done.
- UNIX memory management scheme includes swapping and demand paging.

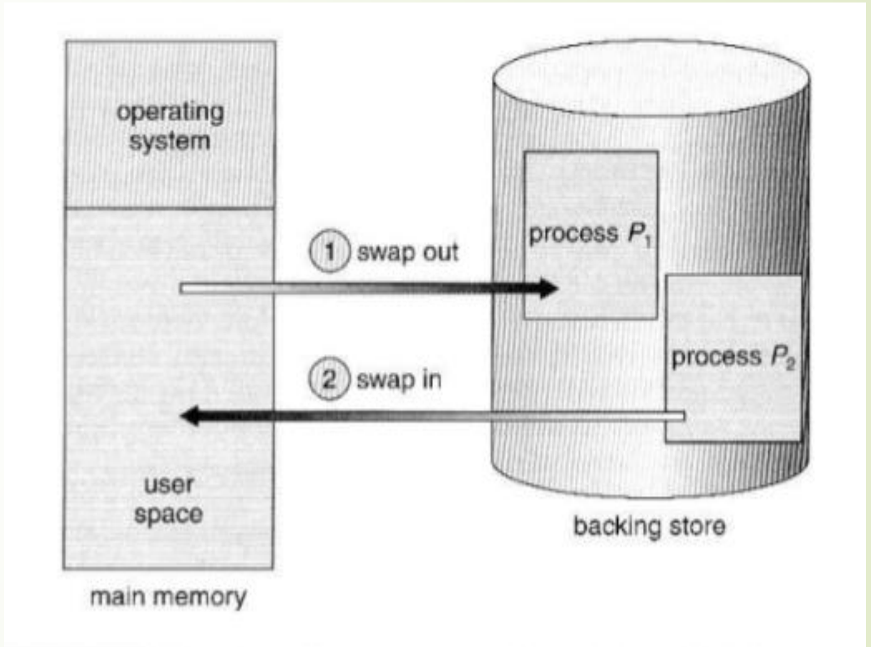
UNIX Memory Management Strategies

- Overlays
  - Program will be place into memory during execution.
  - However, a large program will divide into small pieces and loading the pieces as they needed.
  - Overlays will replace the new pieces with the program which is unused.
  - UNIX is using this technique to run a new program by fork the running process which is also known as fork-exec.

- Swapping
  - Swapping consists of bringing in each process in physical memory entirely and running it.
  - When the process is no longer in use, the process will be terminated or is swapped out to disk.

Swapping•

- The process of moving some pages out of main memory and moving others in, is called swapping.
- A page fault occurs when the CPU tries to access a page that is not in main memory, thus forcing the CPU to wait for the page to be swapped in.
- Since moving data to and from disks takes a significant amount of time, the goal of the memory manager is to minimize the number of page faults.



Swapping of two processes using a disk as a backing store.

Virtual Memory

- UNIX operating system allows user to fully utilize the physical memory installed in the system as well as part of the hard disk called swap space which have been designated for use by the kernel while the physical memory is insufficient to handle the tasks.
- Virtual memory managers will create a virtual address space in secondary memory (hard disk) and it will determine the part of address space to be loaded into physical memory at any given time.
- The benefit of virtual memory relies on separation of logical and physical memory.

Demand Paging

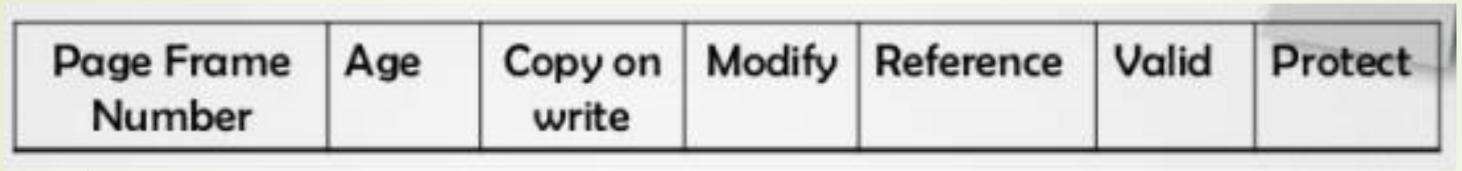
- Paging is a memory allocation strategy by transferring a fixed-sized unit of the virtual address space called virtual page whenever the page is needed to execute a program.

- As the size of frames and pages are the same, any logical page can be placed in any physical frame of memory.

- Every processes will be logical divided and allocate in the virtual address space.

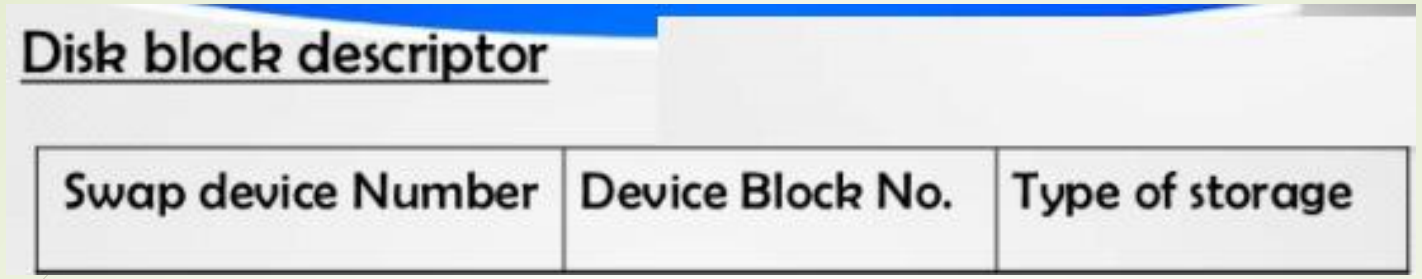
- There is a page table in the virtual memory to allocate and keep tracking of the pages to map into the frames.

Page Table



Page Frame Number	Age	Copy on write	Modify	Reference	Valid	Protect

- frame # contains the physical frame where the virtual page is stored
- age is processor dependent, and is meant to maintain how long it has been since the page was accessed.
- Copy on Write store the copy on write bit, which is used in UNIX systems to, among other things, render fork efficient.
- Modify is a single bit that indicates whether a page has been modified since last swapped in .
- Ref. contains the usage information necessary for a CLOCK- style algorithm.
- Valid is the standard UNIX jargon for resident. A valid page is in main memory, an invalid one is swapped out.
- Protect contains the permission information for the page and is also hardware dependent.

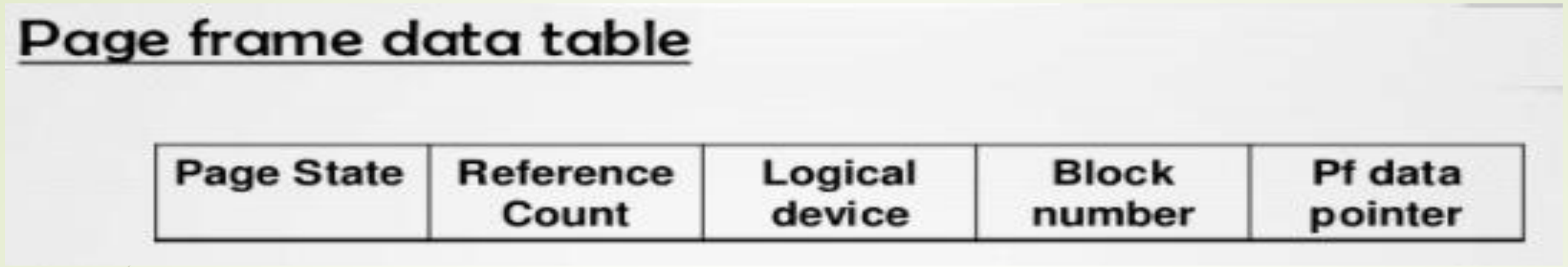


Disk block descriptor

Swap device Number	Device Block No.	Type of storage

- The disk block descriptor contains the information mapping a virtual page to a spot on disk.
- The OS maintains a table of descriptors for each process.

- Device : is basically a pointer to the disk that this page was swapped to.
- Block : is the actual block that the page is stored on
- Type specifies whether the page is new or pre-existing. This lets the OS know if it has to clear the frame first.

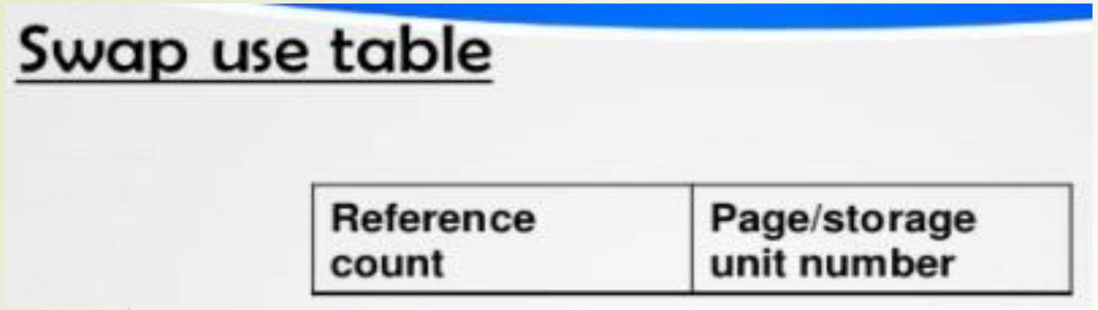


Page frame data table

Page State	Reference Count	Logical device	Block number	Pf data pointer

- The page frame data table holds information about each physical frame of memory (indexed by frame number)
- This table is of primary importance for the replacement algorithm.

- Page state indicates whether or not the frame is available or has an associated page.
- Ref. Count holds the number of processes that refer to this page
- Logical device contains the device number of the disk that holds a physical copy of the page.
- Block # holds the block number on that disk where the page data is located.
- Pf data pointer is a pointer field that is used to thread a singly-linked list of free frames through the frame table.

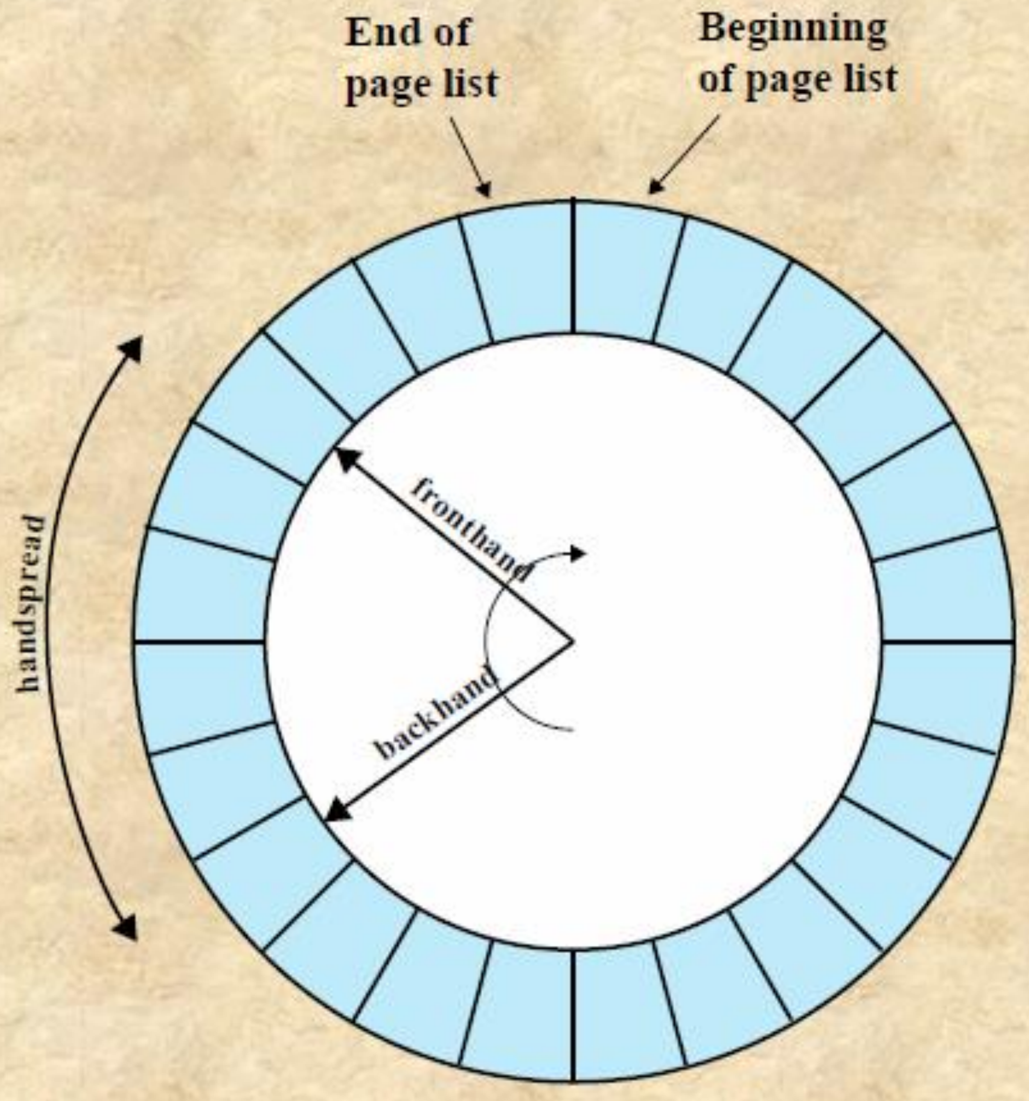

Swap use table

Reference count	Page/storage unit number

- Reference Count : Number of page table entries that point to a page on the swap device.
- Page/storage unit number : Page identifier on storage unit .

PAGE REPLACEMENT

- The page frame data table is used for page replacement
- Pointers are used to create lists within the table
  - all available frames are linked together in a list of free frames available for bringing in pages
  - when the number of available frames drops below a certain threshold, the kernel will steal a number of frames to compensate



Two-Handed Clock Page-Replacement Algorithm

System V Release 4 (SVR4)

- SVR4, developed jointly by AT&T and Sun Microsystems, combines features from SVR3, 4.3BSD, Microsoft Xenix System V, and SunOS.
- It was almost a total rewrite Common facilities of the System V kernel and produced a clean, if complex, implementation.

OpenSolaris

- IS an open source computer operating system based on Solaris created by Sun Microsystems.

Illumos

- is a free and open-source Unix operating system.
- It derives from OpenSolaris, which in turn derives from SVR4 UNIX and Berkeley Software Distribution (BSD).
- illumos comprises a kernel, device drivers, system libraries, and utility software for system administration.
- This core forms the basis of several operating system distributions

SVR4 and Solaris use two separate schemes:

- paging system
  - provides a virtual memory capability that allocates page frames in main memory to processes

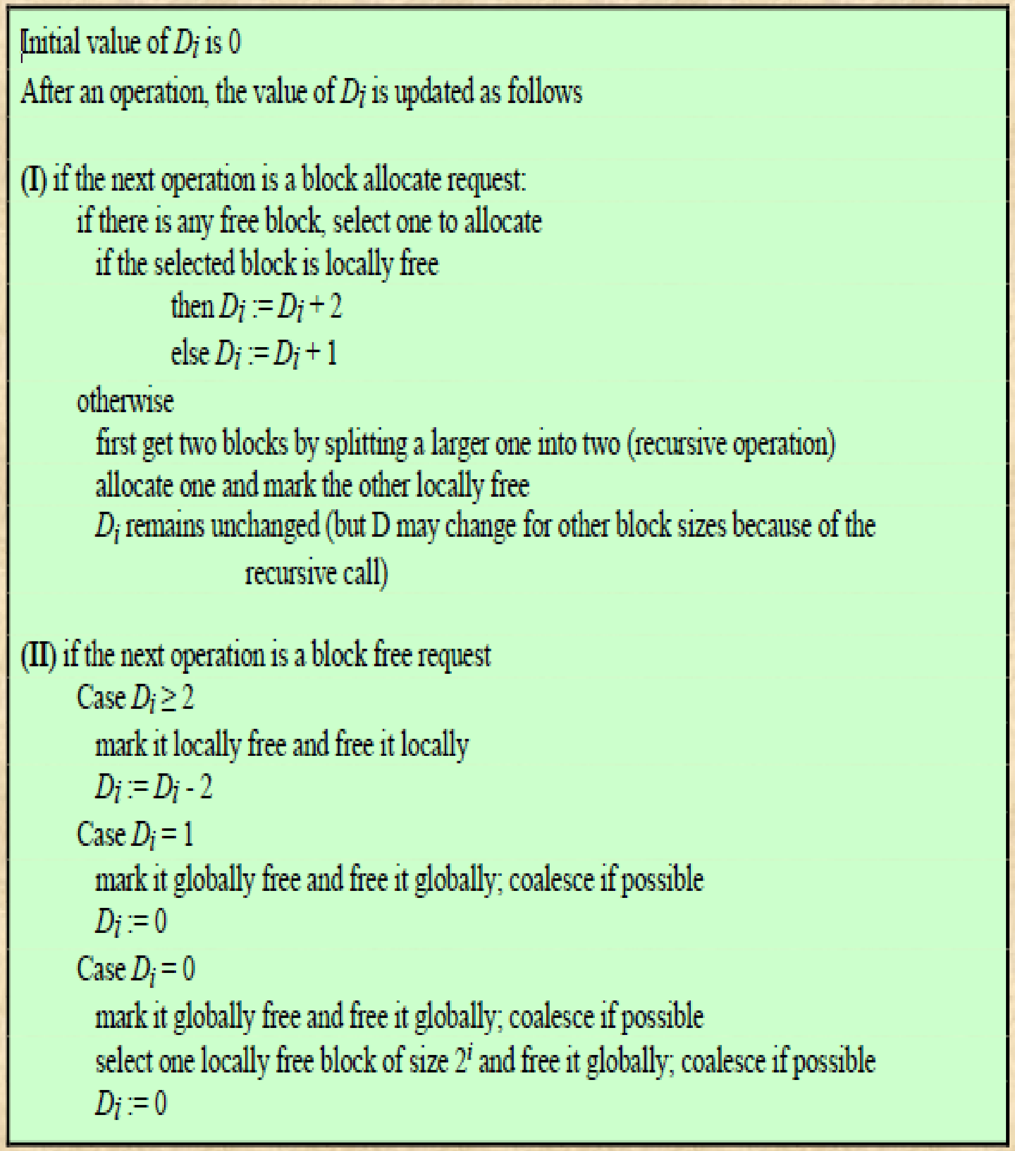
- kernel memory allocator
  - allocates memory for the kernel

Kernel Memory

- UNIX owns a (semi-)private memory space called Kernel memory.
- Kernel uses RAM to keep itself memory resident to ensure that user programs do not overwrite or corrupt the kernel /user's data structures.
- Strong memory protection is implemented in kernel memory management to keep users from corrupting the system area.

LAZY BUDDY

- Technique adopted for SVR4
- UNIX often exhibits steady-state behavior in kernel memory demand
  - i.e. the amount of demand for blocks of a particular size varies slowly in time
- Defers coalescing until it seems likely that it is needed, and then coalesces as many blocks as possible



```
Initial value of $D_i$ is 0
After an operation, the value of $D_i$ is updated as follows

(I) if the next operation is a block allocate request:
      if there is any free block, select one to allocate
        if the selected block is locally free
            then $D_i := D_i + 2$
            else $D_i := D_i + 1$
      otherwise
        first get two blocks by splitting a larger one into two (recursive operation)
        allocate one and mark the other locally free
        $D_i$ remains unchanged (but D may change for other block sizes because of the
                recursive call)

(II) if the next operation is a block free request
      Case $D_i \geq 2$
        mark it locally free and free it locally
        $D_i := D_i - 2$
      Case $D_i = 1$
        mark it globally free and free it globally; coalesce if possible
        $D_i := 0$
      Case $D_i = 0$
        mark it globally free and free it globally; coalesce if possible
        select one locally free block of size $2^i$ and free it globally; coalesce if possible
        $D_i := 0$
```

Lazy Buddy System Algorithm

Conclusion

- Every operating system has different memory management.
- UNIX also has their exclusive memory management strategies to manage the memory resource optimally.
- **nearly UNIX: variable partitioning with no virtual memory scheme**
- **current implementations of UNIX and Solaris make use of paged virtual memory**
- UNIX is using multiple and variable partitioning so that the memory can be stored and use more flexible.
- UNIX also fully utilized the virtual memory (physical memory and swap space) by using demand paging.

REFRENCES

- REF1// http://self.gutenberg.org/articles
- REF2//www.ukessays.com/essays
- Os Internals & design --- William Stallings2.
- The design of the unix operating system --- Maurice J. Bach prentice