***Memory forensics***

1. Memory manger-the part that handles the main memory and transfers between it and the disk.

**Virtual memory**

Motivation-software sizes are increasing faster than memory.

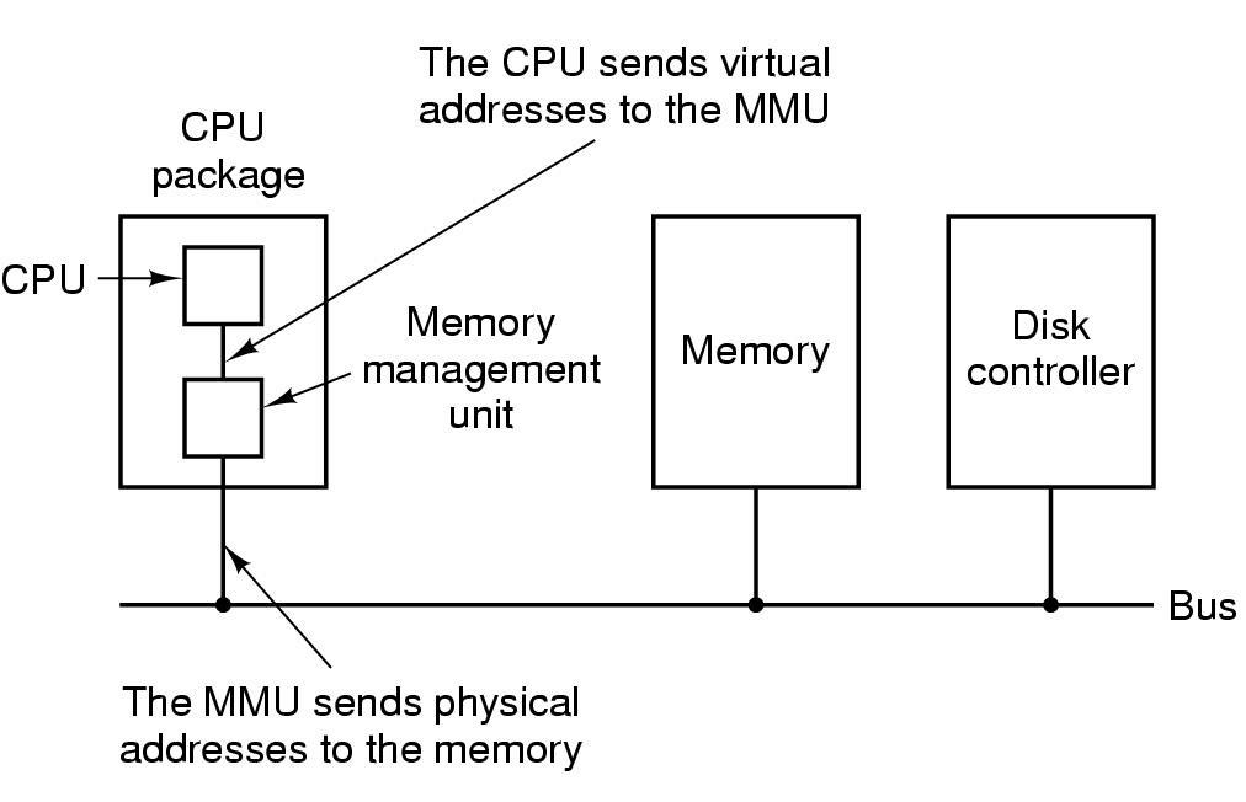
Allows program to run even if the program is partially in the memory.

Each program gets its own address space broken to pages.

-virtual address (generated by the program CPU) is bound to a separate physical address (address seen by the memory unit).

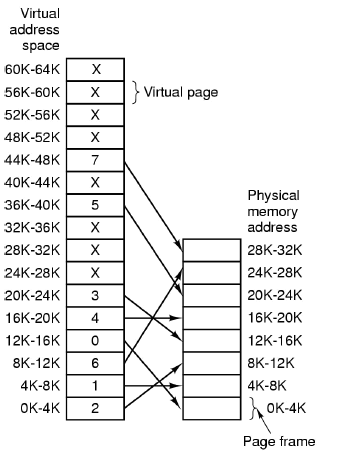
- Virtual address go to Memory Management Unit (MMU) that maps to physical address.

-The user program sees only the virtual addresses.



Paging-partition memory into small equal chunks called pages and physical memory are called frames.

OS maintains a page table for each process (sees only the process’s pages), memory address consists of page number and offset within the page



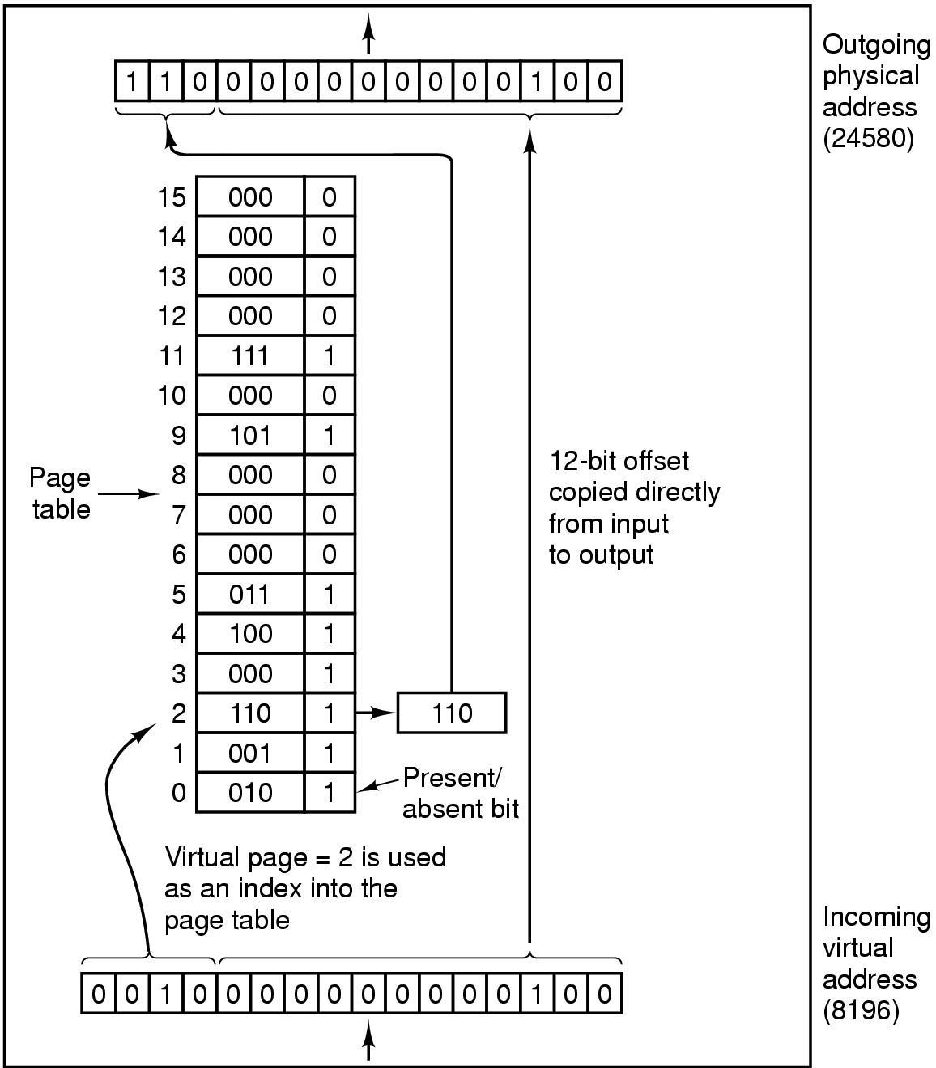
Example:(only 8 pages are mapped to frames because we have only 8 physical page frames)

A Present/absent bit keeps track of which pages are physically present in memory.

MOV REG,8192

Virtual address 8192 is sent to the MMU which transforms to 24576.

Page fault-if the page doesn’t exist there’s a need to evict a little used page frame and write it to disk then fetch the page just referenced into the page frame, changes the map and restarts the instruction.



How it works:

Incoming 16 bit virtual address is split into 4 bit of page number and 12 bit offset so we can address all 4096 bytes within a page.

The page number is used as an index to the into the page table.yielding the number of page frame if the present/absent bit is 0 a trap is caused,if the bit is 1 the page frame number is copied to the high order 3 bits of the output register,12 bit offset is unmodified.

-referenced bit –is to indicate whether a page was referenced

-dirty(modified) bit to indicate whether the page has been altered since it was loaded to main memory.

If no change has been made, the page does not have to be written to the disk when it needs to be swapped out

-the bigger the page, the smaller the table (very important because every process has a table)

Speeding-using tlb (high speed cache).

-to keep the tables small we can make a table of tables, but not every entry must contain a table. we do paging on the table itself

Inverted page table- one entry per page frame in real memory.

\*paging is very slow, it’s much faster to connect to a computer on other network then to get data from disk

2) Memory protection-

Why does memory need to be protected?

no process accesses foreign address, without this a process can change memory belonging to another process or worse it can crash the system by changing memory that belongs to the kernel.

Text segment- read only

Bits used to set permissions on a per (entry) segment basis

\*read write, execute

\*valid, invalid-pages that a process is allowed to access are marked valid, other pages and nonexistent pages are marked invalid

\*\*NX Bit- on newer processors –add bit to page table to prevent execution of non-program data

(Could be malicious code)

Outside of a single process there must be a mechanism to detect which segment belong to which process.

When a process is executing it’s segments are marked valid, segments belonging to other processes are marked as invalid and every attempt to access those segments results in a segmentation fault or bus error and causes the process to be terminated.

Linux kernel sends SIGSEGV or SIGBUS to process and exits immediately.

\*separate mappings allows us to keep programs isolated.

On non-paging system it’s implemented using a segment table, the virtual address along with the permissions defines the behavior using the bits.

\*pages that are read only can be accessed by multiple processes

base and limit registers – the value of the base register is added to a relative address to produce an absolute address. Need to compare to the limit register, if the address is not in bounds an interrupt is generated to the operating system.

**Memory Acuisition**

**3)** Software option:

These tools work by loading a kernel module that maps the desired physical

Addresses into the virtual address space of a task running on the system. At this point,

They can access the data from the virtual address space and write it to the requested

Non-volatile storage. Acquisition software has two ways to make this virtual-to-physical

Address mapping occur:

1) The approach that most, if not all, commercially available tools utilize involves

using an operating system API to create a page table entry

2)approach uses other OS APIs to allocate an empty page table

entry and manually encode the desired physical page into the page table entry

**hardware option:**

Direct Memory

Access (DMA) using a technology such as Firewire, Thunderbolt, ExpressCard, or PCI.

Many disadvantages like:high cost,, Firewire only permits acquisition of the first 4 GB

**memory acquisition should be collected in order of from the most volatile to the least .**

**First of all dump an image for example with the winpmem.**

**Traditional tools wrote out a raw format image in this format** the physical address space is written byte for byte directly into the image file every byte in the file corresponds to the same address in physical memory.

copying the contents of volatile memory to non-volatile storage

“imaging memory” is a sampling of the state of physical memory at a given point in time.

acquisition of discrete units of memory (pages),

**After that analyze the image with a forensic tool like volatility**

**4+5)** anti-forensic

attacks fall into two broad categories – those techniques

which prevent evidence from being acquired, and those

techniques which remove data from the collected evidence

such that the collected evidence can not be suitably analyzed.

Substitution attacks,

in which data fabricated by the attacker is substituted

in place of valid data during the acquisition process

Alternatively

a rootkit might disrupt the acquisition process

altogether (e.g. hang the hardware) when detecting the presence of a forensic agent. This approach is especially

effective against memory acquisition, since the volatility of

the evidence does not permit the investigator to reacquire

the memory under the same conditions.

rootkit’s task is to remain hidden,

causing as little noticeable interruption to normal system

activity. However, once the rootkit detects a memory

acquisition tool is running, it requires that acquisition to be

thwarted or redacted.

ddefy tool

(Bilby, 2006). This tool hooks the physical memory device

and filters certain pages from being read through this

interface, providing instead a cached copy (prior to kernel

modification).

even hardware-based acquisition can be

defeated using very low level manipulation of the memory

controller’s hardware registers

memory acquisition tools do not appear to

implement mechanisms to protect their operations against

anti-forensic attacks

**6)**

**a) Download winpmem-1.4**

**b) Open command prompt as Administrator and navigate to the winpmem folder.**

**c) Write in cmd winpmem\_1.4.exe filename for example memo.dmp**

**d) Download Volatility 2.4 framework**

**e) In the command prompt console type volatility.exe -f imagepath imageinfo**

**to obtain the profile of the image**

**f)** run volatility.exe --profile=**Win7SP1x86 –f imagepath pslist to obtain process list from image**

**g)** run volatility.exe **--profile=Win7SP1x86 -f mem.dmp -p 2912 memdump -D memory/**

**this will extract process 2912 (chrome) to a separate file called 2912.dmp in memory directory.**

**h)** run command **strings 2912.dmp > file.txt**

**this will dump all strings to the file.txt**

**7)The data structures accessed by my program are:**

**The EPROCESS struct which is the process struct,and the PHANDLE\_TABLE (fd’s table).**