CS 451 / 551 / ECE 541

ADVANCED
COMPUTER ARCHITECTURE

SESSION no. 16

University of Idaho

CACHE VS VIRTUAL MEMORY - CONTEXT

VIRTUAL ADDR SPACE >> MAIN MEM.

VM EXTENDS MAIN MEM ONTO DISK

Q: WHAT ELSE IS ON DISK?

A: FILES! MUCH MORE SPACE THAN VM.

Q: IS THE FILE SYSTEM PART OF THE ADDRESS SPACE?

A: NO. ACCESSED BY NAME

DISK HAS A DIRECTORY

NAME → BLOCK NUMBER →

TRACKS & SECTORS

"NAME SPACE"

Parameter	First-level cache	Virtual memory
Block (page) size	16–128 bytes	4096–65,536 bytes
Hit time	1–3 clock cycles	100–200 clock cycles
Miss penalty	8–200 clock cycles	1,000,000–10,000,000 clock cycles
(access time)	(6–160 clock cycles)	(800,000–8,000,000 clock cycles)
(transfer time)	(2–40 clock cycles)	(200,000–2,000,000 clock cycles)
Miss rate	0.1–10%	0.00001–0.001%
Address mapping	25–45-bit physical address to 14–20-bit cache address	32–64-bit virtual address to 25–45-bit physical address

Figure B.20 Typical ranges of parameters for caches and virtual memory. Virtual memory parameters represent increases of 10 to 1,000,000 times over cache parameters. Normally, first-level caches contain at most 1 MB of data, whereas physical memory contains 256 MB to 1 TB.

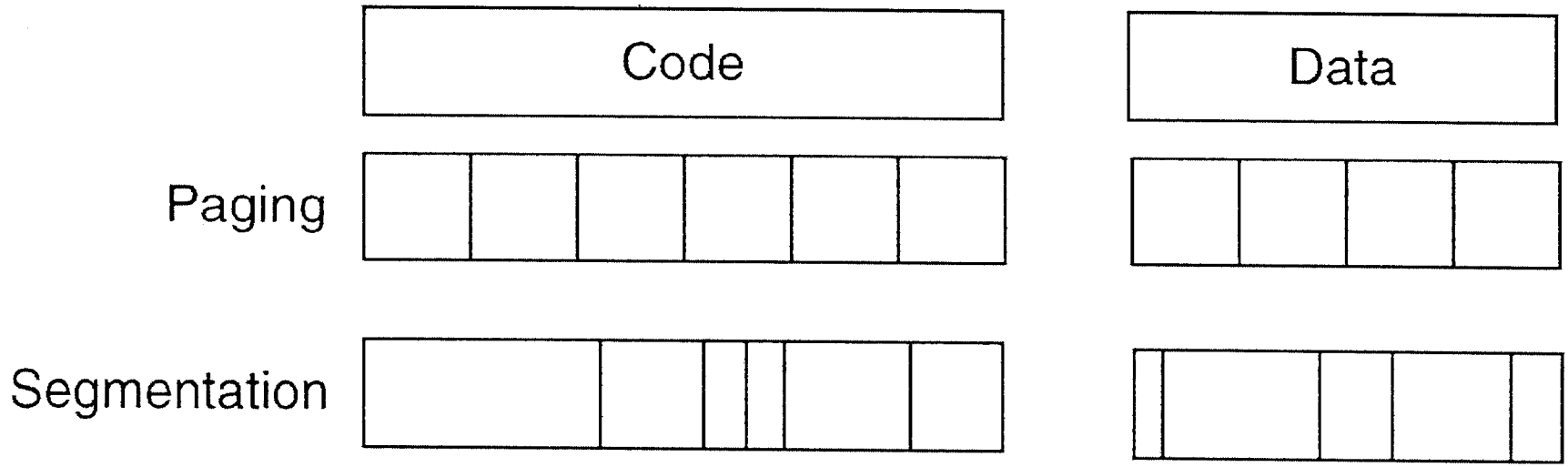
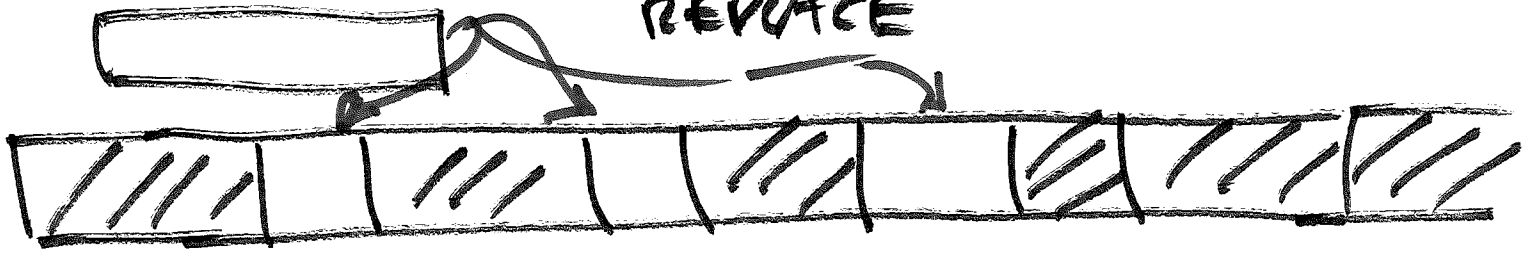


Figure B.21 Example of how paging and segmentation divide a program.

PAGES VS SEGMENTS

- FIXED SIZE
- VARIABLE SIZE

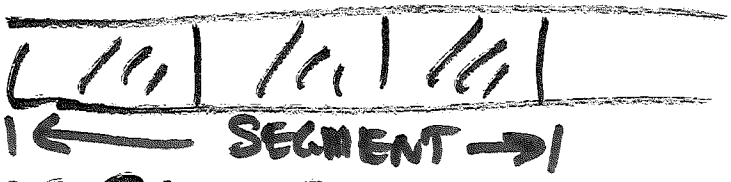
- CHOICE AFFECTS H/W & S/W
- SEGMENTS - THEORETICALLY MORE EFFICIENT USE OF SPACE.



EXTERNAL FRAGMENTATION

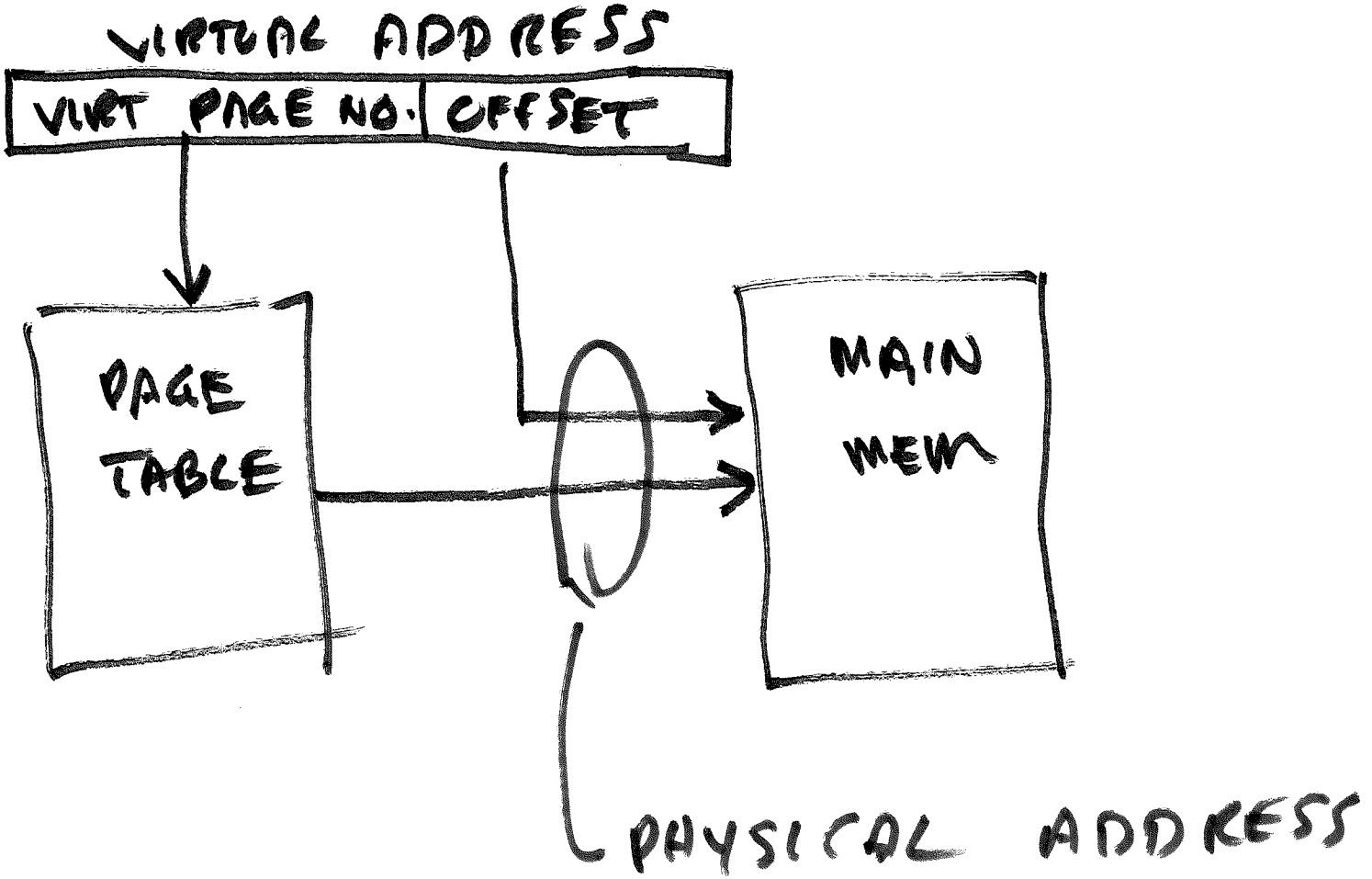
COMPROMIZE

- MIN BLOCK SIZE



SEGMENT = STRING OF BLOCKS

PAGE TABLES


VIRTUAL ADDRESS
VIRT PAGE NO.
OFFSET
PAGE TABLE
MAIN MEM
PHYSICAL ADDRESS

PAGE TABLE IN S/W

IS IT IN MEMORY?

YES - GENERATE ADDR, ACCESS DATA

NO - GO FIND IT!

BRING IT INTO MEM.

REPLACE OLD PAGE BUT...

IF OLD PAGE IS "DIRTY"

WRITE OUT OLD PAGE

COMPLEX - BUT DON'T DO IT OFTEN!

SEARCH — HOW?

LINEAR $T(n) = \Theta(n)$

BINARY $T(n) = \Theta(\log n)$

HASH $T(n) = \Theta(1)$

REMEMBER — GOING THROUGH CACH

BUILD SPECIAL CACHE

HARDWARE SUPPORT

TRANSLATION LOOKASIDE BUFFER

STEPS:

1. IS THE ~~BLOCK~~ PAGE IN MEMORY?

   S/W - HASH FN

   H/W - CONTENT ADDRESSABLE MEMORY

(2) IS IT LEGITIMATE?
VALID, ACCESS PERMISSION

(3) SELECT ADDRESS TO OUTPUT

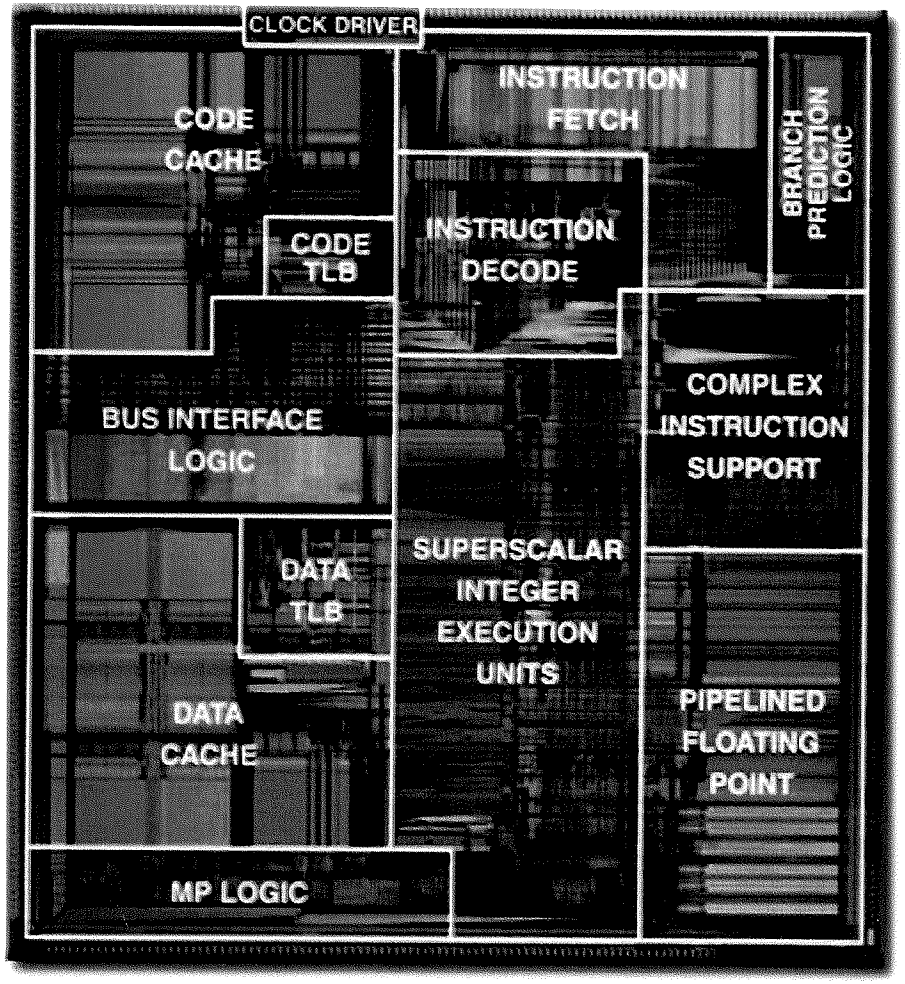
(4) CONCATENATE HIGH & LOW ORDER
BITS - FORM PHYSICAL ADDRESS

EDN MAG ~400 COMMON PROCESSORS
AVAILABLE

- MOST EMBEDDED -
MOST DON'T NEED CACHE,
MEMORY MANAGEMENT,

Intel Pentium Processing Unit

A High Performance "Uniprocessor"



Software environment:
C, C++, Java, 4GLs,...

Operating system support
- CPU Scheduling
- Memory Management
- Input/Output
- Multithreading

Gregory W. Donohoe

University of Idaho
College of Engineering

V = VALID R/W = READ OR WRITE

U/S = USER/SYSTEM

D = DIRTY A = ACCESSED RECENTLY?

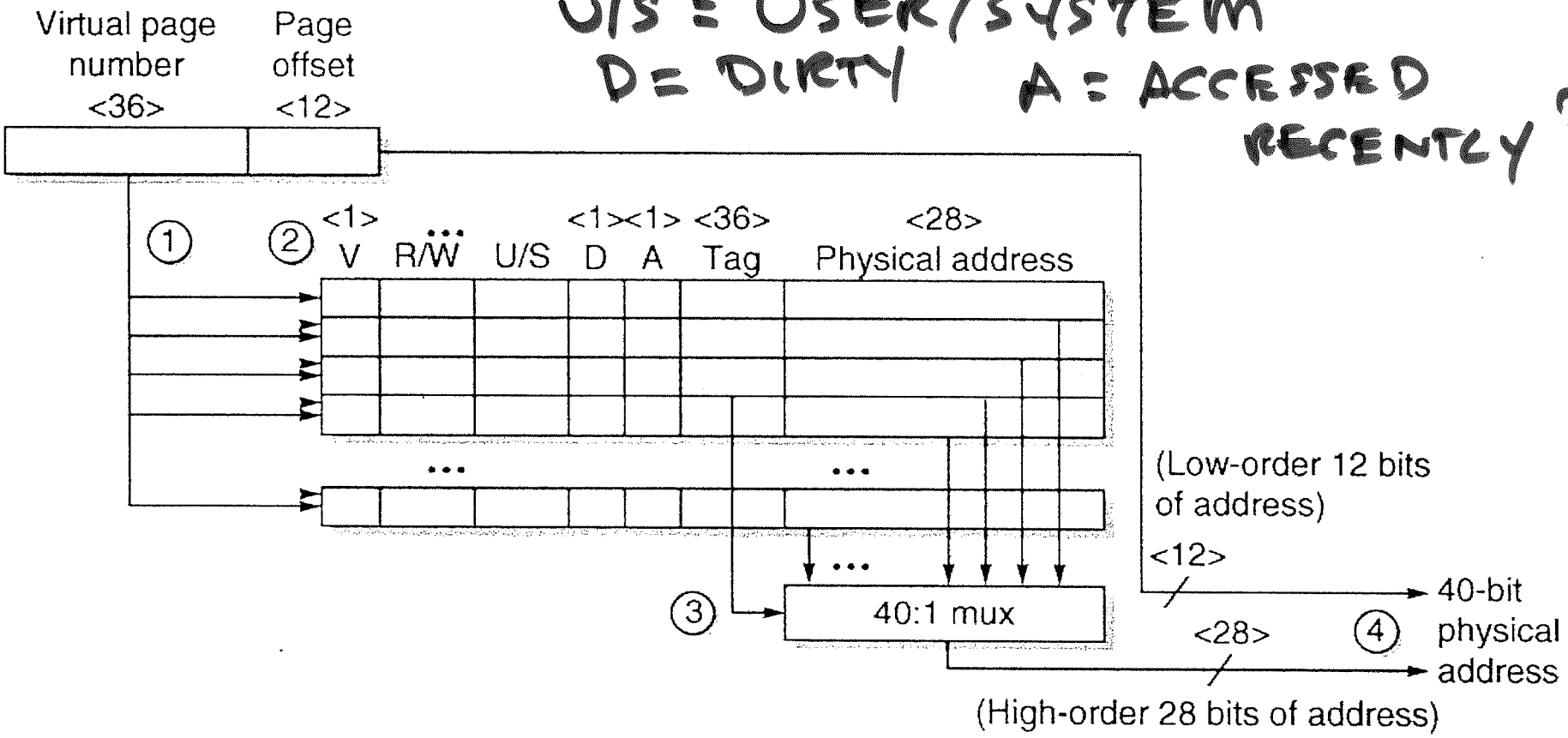
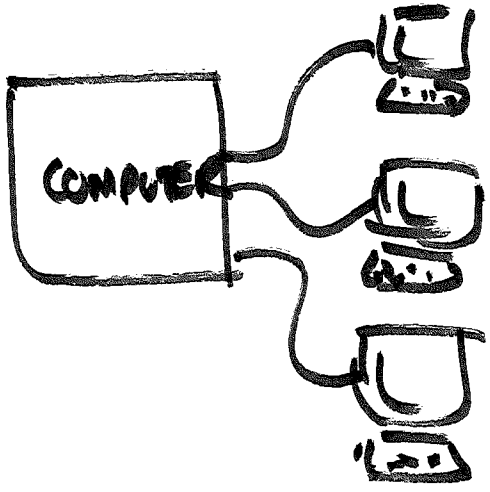


Figure B.24 Operation of the Opteron data TLB during address translation. The four steps of a TLB hit are shown as circled numbers. This TLB has 40 entries. Section B.5 describes the various protection and access fields of an Opteron page table entry.

B.S. PROTECTION
2nd BIG ASPECT OF MEMORY MANAGEMENT

OLD DAYS - ONE PROCESS
"BATCH" PROCESSING - "JOBS"

TIME SHARING USERS



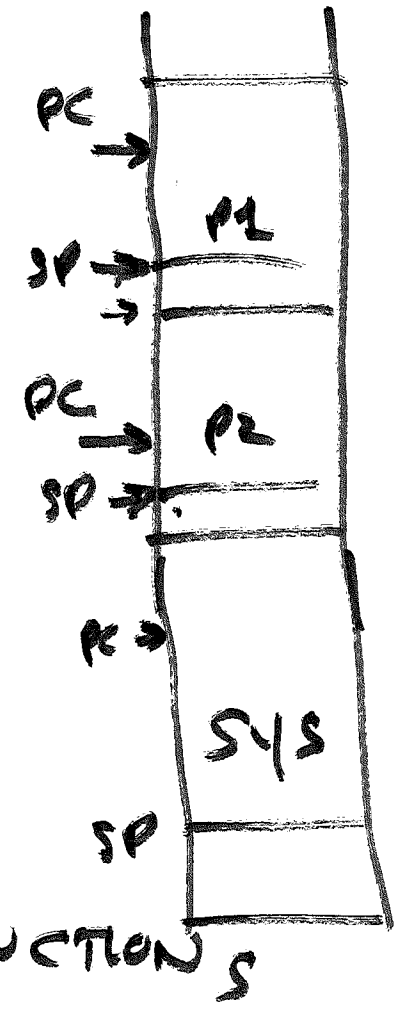
- APPEARED SIMULTANEOUS TO USERS
- REALLY "MULTITASKING"
- RAPIDLY SWITCHING BETWEEN USERS

University of Idaho

USER = PROCESS ABSTRACTION

PROCESS CONTEXT:

- PC
- STACK
- STATUS (NEVC)
- PRIVILEDGE
  - ~~USER (KERNEL)~~
  - USER
  - { SUPERVISOR (KERNEL)
    SYSTEM



CONTEXT SWITCH:

- INTERRUPT - BETWEEN INSTRUCTIONS
  - TIMER -
  - I/O REQ
  - SYSTEM

University of Idaho

MULTIPLE PROCESSES IN MEMORY AT ONCE

- PROCESS: VIRTUAL ADDRESSES
- IN MEM: PHYSICAL ADDRESSES

THREAD?

THREAD RESIDES IN MEMORY SPACE OF PROCESSES

THREADS ~~SHARE~~ MEMORY AND SOME CONTEXT

LIGHT-WEIGHT PROCESS (LWP)

~~SWT~~ SWITCHES QUICKLY

EACH PROCESS

- CAN BE INTERRUPTED AT ANY TIME (BETWEEN INSTRUCTIONS)
- MUST RESUME AS IF NOT INTERRUPTED

PROTECTION (MEMORY)

1) KEEP PROCESSES FROM INTERFERING (OVERWRITING MEMORY FROM ANOTHER PROCESS.)
   - EACH HAS OWN PHYSICAL ADDRESS SPACE.

2) SECURITY - KEEP (PREVENT) EXTERNAL AGENTS FROM MODIFYING RUNNING SYSTEM.