Thursday, October 28, 2021 11:06 AM

Memory Management ---

Memory --------- RAM memory

Kernel space /user space

Volatile ? Data is present only till power on !!!!!

Memory location = 1 byte = its an address= RAM address

Physical address

Logical address, offset address, base address, actual address!!!!!!!!!!!!!

RAM

Ram location/address PHYSICAL address	Data (each location has one byte of data)
1 - base address of p1	P1 0 I1
2	1 I2
3	2 D1
4	3 D2
5	4 D2
6 - base address of p2	P2 0 I1
7	1 I2
8 actual address	2 I3---------- actual address of I3 **base + offset** 6 + 2 =8
9	3 I4
10	4 D1
11	5
12	6 D2
13	
14	
15	
.....	

Process P1

Offset address / logical address	
0	Instr
1	Instr
2	Data
3	Data
4	data

Process P2

0	I1
1	I2
2	I3
3	I4
4	D1
5	

6	D2

High Level Schedular , LTS = Long Term Schedular = this is the kernel module that decides which process can be loaded in the RAM

2 Schemes to load processes into RAM

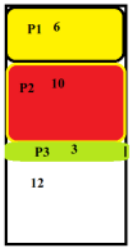
1. Variable partition scheme
2. Fixed partition scheme

What is getting partitioned ? RAM

How many number of partitions ?
What will be the size of partitions ?
Will the size and number of partitions change ?

Questions	Variable partition scheme	Fixed partition scheme
How many number of partitions	changes as new processes come and old processes finish	pre divided into fixed number of partitions = **FRAME**
What will be the size of partition	Varies with size of processes, different partitions have different sizes	the size of each FRAME is fixed and EQUAL
Will the size and number of partitions change	always	NO

Variable Partition Scheme

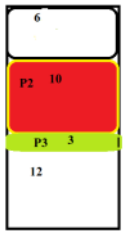


The RAM has 4 partitions

P1 , P2, P3 , Free hole

All have different sizes
This will change if new process comes or any process terminate

PROBLEM ----------- External Fragmentation



A new process arrives
P4 = size of P4 is 14

LTS ? allow or not
Total Free Space = 12+6= 18

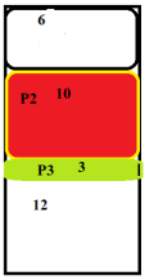
but still LTS will not allow P4 to load WHY ?
Free space is not consecutive/contiguous

FREE SPACE HAI but P4 not allowed

EXTERNAL FRAGMENTATION

Which hole should be used is decided as per First Fit , Best Fit or Worst Fit policy ?

Process P5 arrives --
P5 needs 5 units of space



Process P5 arrives --
P5 needs 5 units of space

LTS ? will allow or not
YES -- contiguous free space is available

if allowed then in which free hole?
depends which policy is used by the LTS
!!!

BEST FIT = select the hole such that
(hole_size - process_size) = minimum

FIRST FIT = select the first hole such that
(hole_size - process_size >= 0)

WORST FIT = select the hole such that
(hole_size - process_size = maximim)

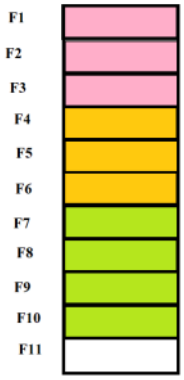
Best fit leaves us with very tiny unusable fragments of free space !!!
Worst fit is more popular .

Actual address calculation =======

Actual location of Instruction or DATA ===== BASE-address + Offset-address

---

Fixed partition scheme = number of partitions are fixed and their sizes are equal



How many partitions = 11 , each of size 4

Each Frame = 4

List of allocated Frames
F1, F2, F3

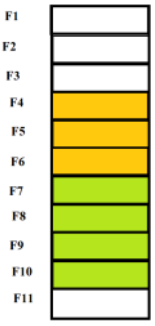
List of Free Frames F11

If a new process P1 = 12

Free consecutive frames are
allocated = F4,F5,F6 are
allocated

if a new process P2 = 15
F7,F8,F9,F10

PROBLEM --- if consecutive free frames are not available = EXTERNAL FRAGMENTATION



Free frames = 4 ,
F1 F2 F3 F11

Total free space = 16

New process P3 = 13

LTS allow ??
P3 needs 4 frames
consecutive

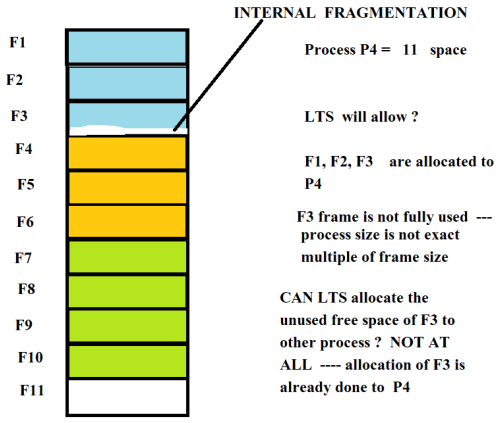
we dont have consecutive
frames
So P3 NOT ALLOWED

External Fragmentation

PROBLEM --- space inside the frame may be wasted if process space is not exact multiple of frame size =INTERNAL FRAGMENTATION

External fragmentation=> consecutive/contiguous space not available==> occures in both fixed partition as well as variable partition schema

Internal Fragmentation=> when size of process is not exact multiple of size of frame so there is chance of last frame may not get fully occupied



First FIT , Best Fit , Worst Fit }}} can be used to decide which frames must be allocated to the process!!!!

The entire process was loaded in the RAM + It MUST LOADED in CONSECUTIVE LOCATION

main reason to external fragmentation

We Were facing problem of EXTERNAL FRAGMENTATION !!!!!!!

TWO SOLUTIONS to external fragmentation

1. COMPACTION = shift the free holes to one end of RAM --- we get contiguous free RAM ---- Difficult to implement
   THEORETICAL

2. PRACTICAL solution == Use Segmentation and PAGING

5mins Break

12:40 !!!

TO AVOID EXTERNAL FRAGMENTATION ----------

We will load the entire process RAM BUT we will not insist on CONSECUTIVE LOCATIONs !!!

If the locations are not consecutive STILL the process will be loaded in the RAM

1. VARIABLE PARTITION SCHEME ---without consecutive storage
   SEGMENTATION !!!!

The process is DIVIDED into SEGMENTS !!! On the basis CONTENT !!!

CODE content ---------- CODE SEGMENT
GLOBAL DATA CONTENT ---------- DATA SEGMENT
LOCAL DATA CONTENT ----------STACK SEGMENT
DYNAMIC DATA CONTENT ---------- HEAP SEGMENT

Are all the partitions ----segments are of equals lengths ?? UNEQUAL

Process P1	
S1	0 (**segment offset**) I1
	1 I2
	2 I3
	3 I4
S2	0 V1
	1 V2

S3	0 L1
	1
	2
	3
S4	0 D1

The four segments of P1 can be stored in any location in the RAM
The hole_size - segment_size >=0

The entire process need not be consecutive !!
So we need to keep track of which segment is stored where for EACH process ?
Kernel will use a data structure ---- SEGMENT TABLE (one per process)

SegNo	Segment base address	Segment size
1	**10**	**4**
2	**45**	**2**
3	**55**	**4**
4	**39**	**1**

Now EXTERNAL FRAGMENTATION PROBLEM IS SOLVED TOTALLY ?
NO , but it is REDUCED , but it may still occur if consecutive space is not available for the segment !!!!

ACTUAL ADDRESS CALCULATION ----------

OFFSET ADDRESS IS MADE OF TWO PARTS
1. Seg number
2. Seg offset

Kernel gets address of next instruction in the PC register = **Seg 2 ,offset 1**

1. **Find the base address of seg 2 in segment table = 45**
2. **Add segment offset**
   actual address = 45 +1 = 46 !!!

If the offset of the segment is larger than seg size ------- TRAP interrupt !!!!

5.30 Linux commands session !!!
