disk updation

28th march

Vaxiable		Value		
p		43		
h		13		
m		29		
c		49		
h		36		

h = h*2

C = C+17

on disk write

h = 26, C = 66

if power fails in middle of execution entire program is executed again.

if power fails in middle of variable updation. start variable updation again.

on disk write

h=26

c=66

done.

done written:

update variables

done not written:

execute the operations again.

t	19
d	94
m	36

Variable	Valid	Value	shadow
t	0	19	41
d	1	26	94
m	1	49	36

0=actual value valid
1=shadow valid

Variable	valid	Value	shadow
p	0	43	39
h	~~1~~ ~~0~~ 1	~~13~~ 84	~~42~~ 84
m	1	29	93
c	~~0~~ 1	49	~~19~~ ~~66~~ 66
h	1	26	42

on disk write

h = 84

c = 66

done.

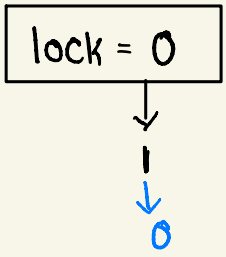
if power fails before "c" shadow updation

then update starts again,

shadow of h is also updated.

but during power cut, h = 84, c = 49 are values.

Variable	valid	Value	shadow
p	0	43	39
h	~~0~~ 0	~~23~~ 46	~~31~~ 46
m	0	29	93
c	~~0~~ 0	~~42~~ 66	~~36~~ 66
h	0	36	42



on disk write

h = 46

c = 66

done.

if (lock == valid == 1)

(value) might be wrong.

after done is written:

update in shadow

make lock 1

make valid of h and c → 1

transfer (shadow → value) of h

make valid = 0

transfer (shadow → value) of c

make valid = 0

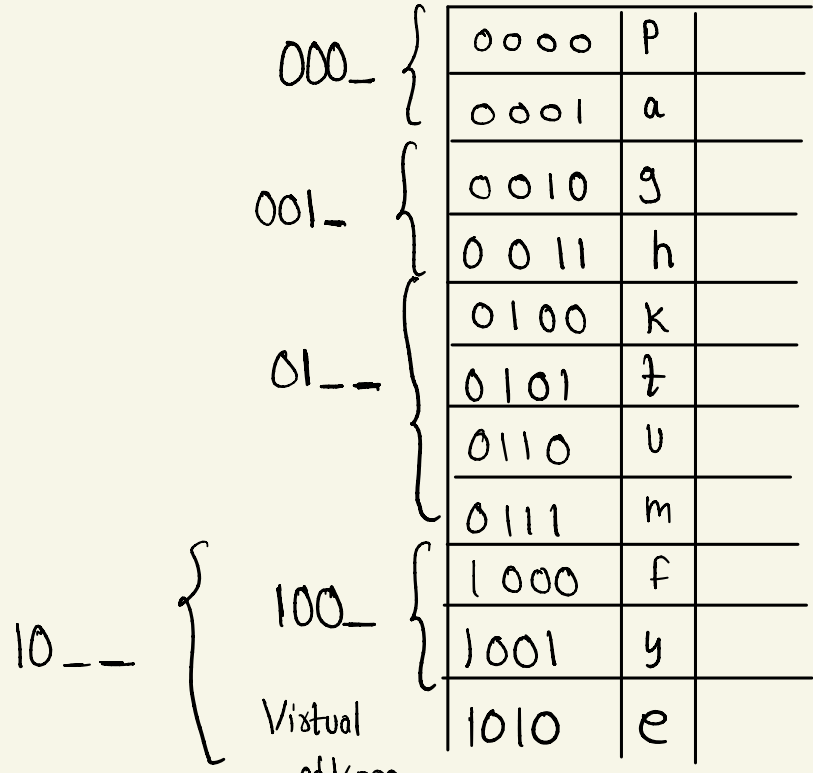
lock = 0

Buddy

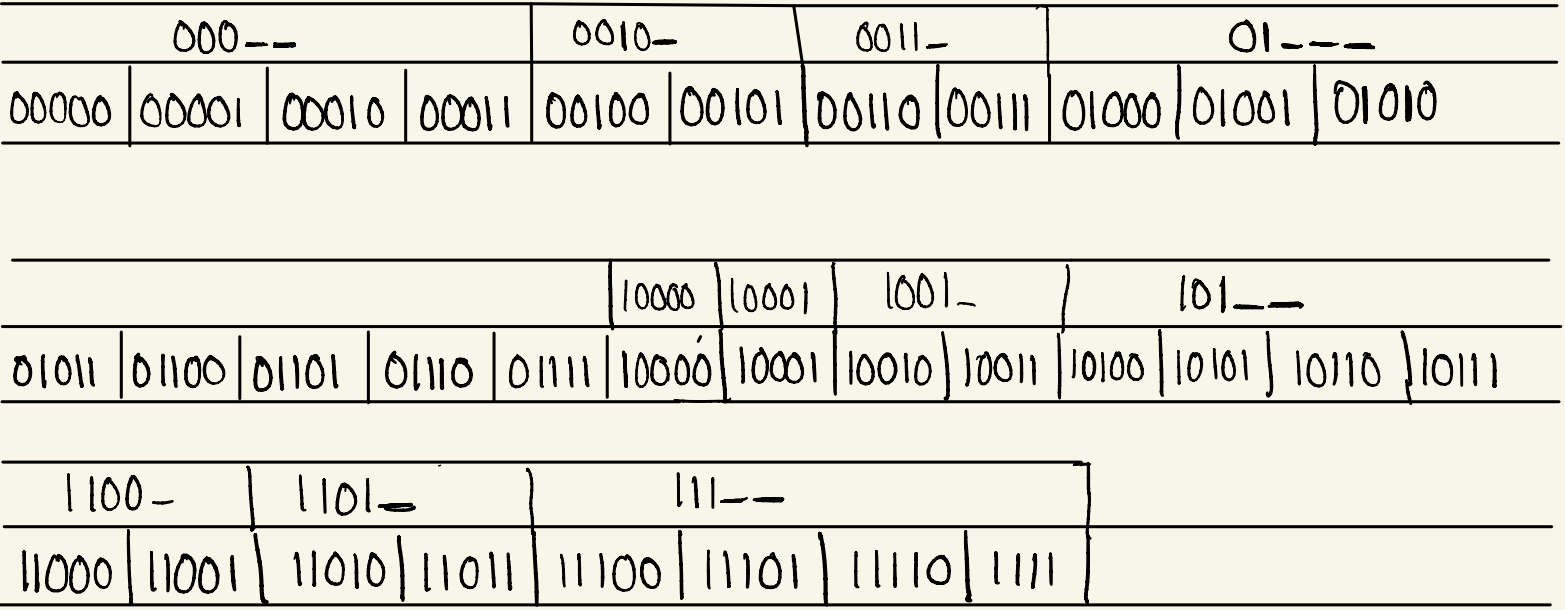
3rd April

Size : 2^k

int p, a, g, h, k, l, u, m, f, y



BUDDY



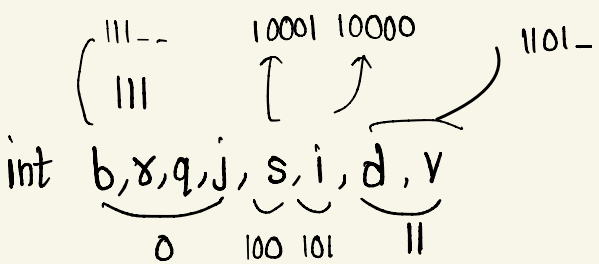
Physical address.

000	001	01	100
1001	0011	101	0010

← virtual buddy

Segment table: 1001001110100 10

virtual	0101	0011
physical	10101	00111



000
001
010
011
100
101
110
111

Segment table: 111, 10001, 10000, 1101

virtual 011 110

physical 111 11 11010

virtual buddy

000	001	01	10
1001	0011	101	000

(4) (4) (8) (8) ✗

(2) (2) (4) (4) ✗

virtual buddy

virtual size (45)

000 , 001 , 01, 10

(8) (8) (16) (16) opqx

abcde fghij klmn

Physical size:

minimum wastage = 3

Segment table: a b c d e f q h i j k l m n o p q x

segment table for 000 is x

01 is x-1

001101

fghij 101

$2x + 2x - 2 = 18$

$x = 5$

011010

klmn 1010

Page table :

a b c d e f g h i j k l m n o p q r s t (20)

Find physical for virtual 0101101

Let pagesize = 8

0101101

page no

10 pages

9 to 16 pages.

Let page size = 32

01 01101

3 or 4 pages

abcde, fghij, klmno, pqrst

fghij 01101

Physical memory = $2^{(5+5)}$

= 1024

Let page size = 16

0101101

number of pages = 5

abcd, efgh, ijkl, mnop, qrst

010 1101

ijkl 1101

Physical memory = 2^8

Segment table : a b c d ef (6)

3 buddies

ab, cd, ef – possible

a, bc, def ✓

2^k 2^{k-1} 2^{k-2}

4 2 1

abc, de, f – ✗

---

a | bcd | efg – ✓

abc, de, f

write smallest sequence so that work is done faster

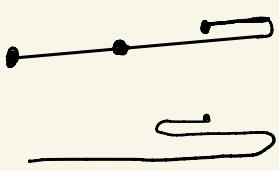
in FCFS. composed SSTF

10
30
40
60

40 60 30 10

40 60 90 110

40 80 50 130



Job	Arrival	Serve
A	0	50
B	10	35

how to serve so that

1) Work over smallest

2) Average wait is smallest.

Job	Arrival	service
A	0	50
B	10	K

for which 'K' preamption reduces average wait.

let swap in 5
swap out 7

$k < 40$

$\frac{k}{2}$

$40/2$

$40/2 > K/2$

RAM	operation	Disk
	Disk (45)	45
19	RAM (19)	
19, 24	RAM (24)	
	Disk (13)	45, 13
19, 24, 17	RAM (17)	
	Disk (42)	45, 13, 42
NIL	RAM (91)	45, 13, 42, 19, 24, 17, 91
71	RAM (71)	

(Max capacity of RAM = 3)

automatic transfer from RAM → Disk

10th April

RAM	operation	Disk
	Disk (12)	12
19,	RAM (19)	
19, 16	RAM (16)	
	Disk (13)	12, 13
	RAM → Disk	12, 13, 19, 16
17	RAM (17)	
	Disk (49)	12, 13, 19, 16, 49
	OVER	12, 13, 19, 16, 49, 17

write operation

RAM	Operation	Disk	Work
		12, 17, 45, 19, 36, 84, 71, 93, 86	
	a = Disk	~~12~~, 17, 45, 19, 36, 84, 71, 93, 86	a = 12
	b = Disk	~~12~~, ~~17~~, 45, 19, 36, 84, 71, 93, 86	b = 17
19, 36	c = RAM	~~12~~, ~~17~~, ~~45, 19, 36~~, 84, 71, 93, 86	c = 45
same	d = Disk	~~12~~, ~~17~~, ~~45, 19, 36~~, ~~84~~, 71, 93, 86	d = 84
36	e = RAM	~~12~~, ~~17~~, ~~45, 19, 36~~, ~~84~~, 71, 93, 86	e = 19
NIL	RAM → Disk	12, 17, 45, 19, 36, 84, 71, 93, 86	
71, 93	f = RAM	~~12~~, ~~17~~, ~~45, 19, 36~~, ~~84, 71, 93~~, 86	f = 84
	RAM → DISK	12, 17, ~~45, 19, 36~~, ~~84~~, 71, 93, 86	

read operation

RAM	operation	Disk
	Disk (45)	45
19	RAM (19)	
19, 24	RAM (24)	
	Disk (13)	45, 13
NIL	RAM (17)	45, 13, 19, 24, 17
	Disk (42)	45, 13, 19, 24, 17, 42
91	RAM (91)	45, 13, 42, 19, 24, 17, 42
91, 71	RAM (71)	same

automatic transfer when RAM reaches max capacity.

write operation

while handling both read and write simultaneously during RAM -> DISK (1st read, 2nd write)

---

during read operation from RAM if write- RAM is not empty then RAM -> D will take place automatically.

RAM- write	RAM - read	operation	Disk	
NIL	NIL		12, 17, 34, 81, 29, 38, 73, 86, 49	
	17, 34	a = RAM	~~12, 17, 34~~, 81, 29, 38, 73, 86, 49	a = 12
		Disk (39)	~~12, 17, 34~~, ~~39~~, 29, 28, 73, 86, 49	
		RAM(18)		
16, 93		RAM(93)		
NIL	NIL	RAM → Disk	~~12, 17~~, ~~16, 93~~, 29, 38, 73, 86, 49	
	38, 73	b = RAM	~~12, 17~~, ~~16, 93~~, ~~29, 38, 73~~, 86, 49	b = 29
46		RAM(46)		
	38,73	c = disk	~~12, 17~~, ~~16, 93~~, 29, ~~38~~, ~~73~~, ~~86~~, 49	c = 86
	49	d = RAM	~~12, 17, 16, 93, 29, 38~~, ~~46~~, ~~86, 49~~	d = 86

RAM	Operation	Disk
pqr	H(pqr)	
	L(gh)	gh
cd	H(ukcd)	ghpqruk
cdit	H(it)	
	L(hm)	ghpqruk hm

RAM size = 5

H - high level operation

L - low level operation (directly edits on disk)

OVER ghpqrukhm cdit

RAM	Operation	Disk
	H(pqr)	pqr
	L(gh)	pqrgh
d	H(ukcd)	pqrghukc
	H(it)	pqrghukc dit
	L(hm)	pqrghukc dithm

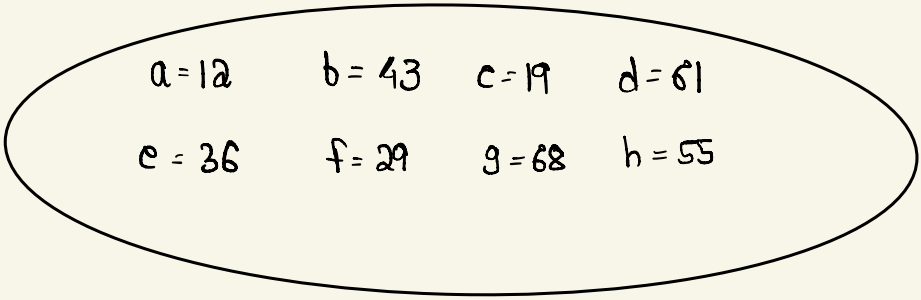
RAM size = 3

write	read	operation	ukhcydmfzkclrd	
		Z(a)	~~u~~khcydmfzkclrd	a=u
		L(ts)	~~u ts~~ cydmfzkclrd	
		Place(7)	~~uts eydm~~ fzkclrd	
		L(a,b)	~~uts eydm ab~~ kclrd	
		Place(4)	~~utse~~ ydmab kclrd	
	mab	Y(bc)	~~utscyd mab~~ kclrd	b=y c=d
		L(gh)	~~utscydmabgh~~ lrd	
	ab	X(d)		d=m
kch		H(kch)		
		Y(e)		

Z (a) = Disk read a=
L(a)= Disk write a
Y(a)=read from ram a=
H(a)=write on ram a
Place 4= take cursor to 4

Cache : Faster than RAM

11th April



{a, f, g} {b, c} {d, e, h}

Cache : U xy V pq W st

U = 0 ⇒ a, f, g not available in cache.

U = 2 ⇒ f in xy.

W = 3 ⇒ h in st

operation	Cache 042017036	
Print (c)	042219036	{19 R → C}
Print (g)	368219036	{68 R → C}
Print (b)	368143036	{43 R → C}
Print (e)	368143236	{36 R → C}
Print (g)	same	no transfer

operation	Cache 029033067	
b = 32	029632067	
Print (e)	029632236	{e R→C}
e = 41	029632741	
Print (c)	029219741	{C R→C b(32) C→R}
Print a	122197 41	12R→C
OVER		{e (41) C→R}

+5 means values to be updated)

P	Q	R	S
a = 29	c = 73	e = 32	g = 86
b = 61	d = 84	f = 61	~~z~~ h = 19

group {PR} {QS}

U xyzw V ijkl

U = 0

U = 1 — 1st variable updated

U = 2 — second variable updated

U = 3 — both updated

U = 4 — neither updated

U = 6
U = 7
U = 8
U = 9

} second page of group

operation	0 4 2 3 9 0 6 1 3 2	
Print (d)	0 4 2 3 9 4 7 3 8 4	7 3 8 4 R → C
Print (e)	9 3 2 6 1 4 7 3 8 4	3 2 6 1 R → C
F = 6 9	8 3 2 6 9 4 7 3 8 4	no transfer
h = 9 2	8 3 2 6 9 7 7 3 9 2	no transfer
Print (a)	4 2 9 6 1 7 7 3 9 2	{3 2 6 9} C → R {2 9 6 1} R → C
Print (c)	4 2 9 6 1 4 7 3 8 4	{7 3 8 4} R → C {8 6 1 9} R → C {8 6 9 2} C → R
c = 3 6	6 3 6 6 1 4 7 3 8 4	
d = 2 9	6 3 6 c 1 ~~8~~ 7 3 2 9 3	
OVER		{73 29}C->R {32,61}R->C {36,61}C->R

	04239 06132	
f = 69	7 4 2 69	
e = 11	8 11 69	
Print(a)		{11 69} C → R {29 61} R → C

f = 69	74269	
Print(a)		{32, 61} R → C {32, 69} C → R {29, 61} R → C