

Only The ContFramePool.C and ContFramePool.H are modified

- 1) All the necessary private variables were added.
- 2) The bit representation is done as 00- allocated

11- free

01- Head of sequence

- 3) I had a hard time using the use of static ContFramePool *head, *tail with which I was thrown into an undefined reference errors but later somehow was able to get along with it. I was able to allocate frames using these static references but while releasing I was going into an undefined references error but if I comment out the infinite for loop in kernel.c , I could see the results as successful.
- 4) A pointer to nFreeFrames is defined to modify it by reference whenever it is to be incremented or decremented
- 5) So I defined an empty constructor ContFramePool(){} to initialise this dummy reference to refer to private member variables and get around with the undefined reference errors which have struggled me all through the code checking
- 6) An overloaded release function(unsigned char*, unsigned long, unsigned long) is added to call it from the static release function using a dummy object to make things simpler
- 7) All the functions from simple frame pool. C were thoroughly reused , modified to make them use 2 bits per frame , 32k frames if 1 bit is used can be bit mapped using one frame, so if 2 bits are used then we can bitmap 16k frames using one frame and this would meet our need.
- 8) All the functions, while loops , if conditionals etc are commented and code is maintained clean to maximum extent.

The screenshot shows a terminal window within the Bochs emulator interface. The terminal output is as follows:

```
Bochs x86-64 emulator, http://bochs.sourceforge.net/
[...]
0000000000001[ ] lt_dilhandle is 0x5594194a73d0
0000000000001[PLUGIN] loaded plugin libbx_biosdev.so
0000000000001[ ] lt_dilhandle is 0x5594194a8bd0
0000000000001[PLUGIN] loaded plugin libbx_parallel.so
0000000000001[ ] lt_dilhandle is 0x5594194a9550
0000000000001[PLUGIN] loaded plugin libbx_exfmrq.so
0000000000001[ ] lt_dilhandle is 0x5594194ab120
0000000000001[PLUGIN] loaded plugin libbx_parallel.so
0000000000001[ ] lt_dilhandle is 0x5594194ac080
0000000000001[PLUGIN] loaded plugin libbx_serial.so
0000000000001[ ] lt_dilhandle is 0x5594194b1180
0000000000001[PLUGIN] loaded plugin libbx_gameport.so
0000000000001[ ] lt_dilhandle is 0x5594194b19b0
0000000000001[PLUGIN] loaded plugin libbx_lodebug.so
0000000000001[ ] reading configuration from bochsrc.bxrc
0000000000001[ ] lt_dilhandle is 0x5594194b2460
0000000000001[PLUGIN] loaded plugin libbx_x.so
0000000000001[ ] installing x module as the Bochs GUI
0000000000001[ ] using log file bochsrc.txt
Next at t=0
(e) [0x0000fffffff0] f000:ffff0 (unk. ctxt): jnpf 0xf000:e05b ; ea5be000
f0
<bochs:1> c
alloc_to_go = 21
alloc_to_go = 20
alloc_to_go = 19
alloc_to_go = 18
alloc_to_go = 17
alloc_to_go = 16
alloc_to_go = 15
alloc_to_go = 14
alloc_to_go = 13
alloc_to_go = 12
alloc_to_go = 11
alloc_to_go = 10
alloc_to_go = 9
alloc_to_go = 8
alloc_to_go = 7
alloc_to_go = 6
alloc_to_go = 5
alloc_to_go = 4
alloc_to_go = 3
alloc_to_go = 2
alloc_to_go = 1
alloc_to_go = 0
Testing is DONE. We will do nothing forever.
Feel free to turn off the machine now.
```

Process and Paper work:

- 1) All the work is done on new development environment.
- 2) I have studied the handout and later went through the kernel.c file to see the desired results and what needs to be performed for the frame pools.
- 3) I had a hard time using the use of static ContFramePool *head, *tail with which I was thrown into an undefined reference errors but later somehow was able to get along with it. I was able to allocate frames using these static references but while releasing I was going into an undefined references error but if I comment out the infinite for loop in kernel.c , I could see the results as successful.

4)

```

cscce410@cscce410-VirtualBox: ~/Documents/MP2_Sources/MP2_Sources$ make
  fno-pgtt _fno-stack-protector -fleading-underscore -fno-asynchronous-unwind-tables -c -o machine.o machine.c
  nasm -f elf -o machine_low.o machine_low.asm
  i386-elf-ld -melf_i386 -T linker.ld -o kernel.bin start.o utils.o \
    kernel.o assert.o console.o \
    cont_frame_pool.o machine_low.o
kernel.o: In function '_main':
kernel.C:(.text+0x3a): undefined reference to '__cxa_guard_acquire'
kernel.C:(.text+0x71): undefined reference to '__cxa_guard_release'
kernel.C:(.text+0x79): undefined reference to '_ZN13ContFramePool4headE'
cont_frame_pool.o: In function '__ZN13ContFramePool7getHeadEv':
cont_frame_pool.C:(.text+0x99c): undefined reference to '_ZN13ContFramePool4headE'
cont_frame_pool.o: In function '__ZN13ContFramePool7setHeadES_':
cont_frame_pool.C:(.text+0x9e7): undefined reference to '_ZN13ContFramePool4headE'
cont_frame_pool.o: In function '__ZN13ContFramePool7setTailES_':
cont_frame_pool.C:(.text+0xa09): undefined reference to '_ZN13ContFramePool4tailE'
make: *** [makefile:47: kernel.bin] Error 1
cscce410@cscce410-VirtualBox:~/Documents/MP2_Sources/MP2_Sources$
```

5)

```

cont_frame_pool.C
56 void test_memory(ContFramePool * _pool, unsigned int _allocs_to_go);
57 /*-----*/
58 /* MAIN ENTRY INTO THE OS */
59 /*-----*/
60 /*-----*/
61 int main() {
62     Console::init();
63
64     /* -- INITIALIZE FRAME POOLS -- */
65
66     /* ---- KERNEL POOL ---- */
67
68     /*---- KERNAL_POOL ----*/
69     ContFramePool kernel_mem_pool(KERNEL_POOL_START_FRAME,
70                                   KERNEL_POOL_SIZE,
71                                   0,
72                                   0);
73
74     /|
75
76
77     ContFramePool::sethead(kernel_mem_pool);
78
79
80
81
82     /* ---- PROCESS POOL ---- */
83
84 }
```

```

cont_frame_pool.C
47 class ContFramePool {
48
49     private:
50         static ContFramePool head;
51         static ContFramePool tail;
52
53     /* -- DEFINE YOUR CONT FRAME POOL DATA STRUCTURE(S) HERE. */
54
55     //static ContFramePool data;
56
57     private:
58
59     unsigned char * bitmap;           // We implement the simple frame pool with a bitmap
60     unsigned long nfreedframes;      // Where does the frame pool start in phys mem?
61     unsigned long base_frame_no;     // Size of the frame pool
62     unsigned long info_frame_no;     // Where do we store the management information?
63     unsigned long n_info_frames;
64
65
66
67     public:
68
69     static ContFramePool gethead(){
70         return (ContFramePool::head);
71     }
72     static ContFramePool gettail(){
73         return (ContFramePool::tail);
74     }
75
76     static void sethead(static ContFramePool x){
77         ContFramePool::head = x;
78     }
79     static void settail(static ContFramePool x){
80         ContFramePool::tail=x;
81     }
82 }
```

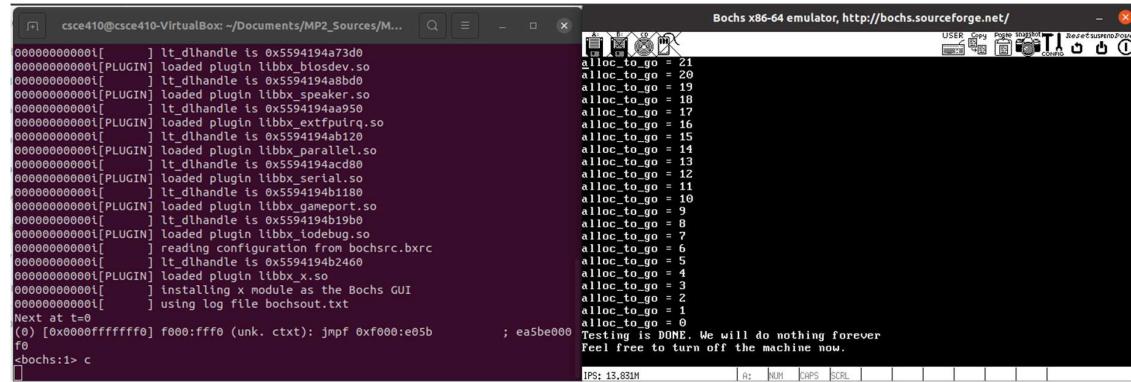
- 5) So tried creating an empty initialiser for ContFramePool class with which the use of static release function has been easier. I used the dummy object to just refer the variables of each of these pool and was able to access them using call by reference and avoided double modifications.
- 6) All the paper work is attached in this document, the code was pushed onto git and I would make sure that I would push more frequently from later on.
- 7) I was trying to figure out why I couldn't access debugger and would try to get it resolved soon with the new dev environment(I'll try reinstalling the new environment again)
- 8) The Zipped folder contains the image of the MP and results along with work done to see if the code actually works on a paper.
- 9) I changed directory to where the files were present
- 10) Later I listed the files using the long listing format ls -l
- 11) Then using make command I compiled the source code
- 12) After compiling I found a new kernel.bin is generated

- 13) To copy the kernel onto an image of a floppy drive I used the command ./copykernel.sh
- 14) Then I emulated using bochs emulator to get the desired output as shown in the following image using bochs -f bochsrc.bxrc

The following is my github work link

https://github.tamu.edu/kausht14/OS_MP2

The screenshot of the result is attached below.



The screenshot shows a terminal window within the Bochs emulator interface. The window title is "Bochs x86-64 emulator, http://bochs.sourceforge.net/". The terminal output displays a log of plugin loading and configuration:

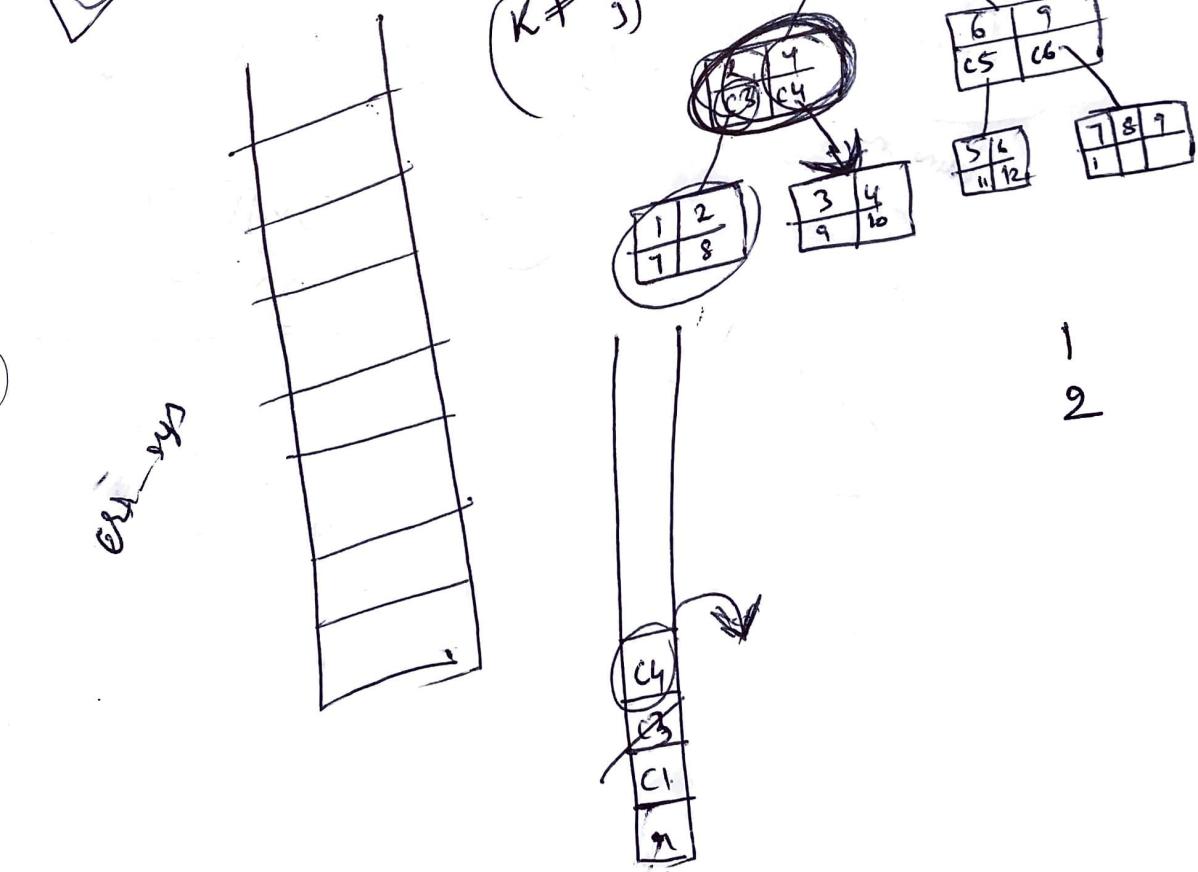
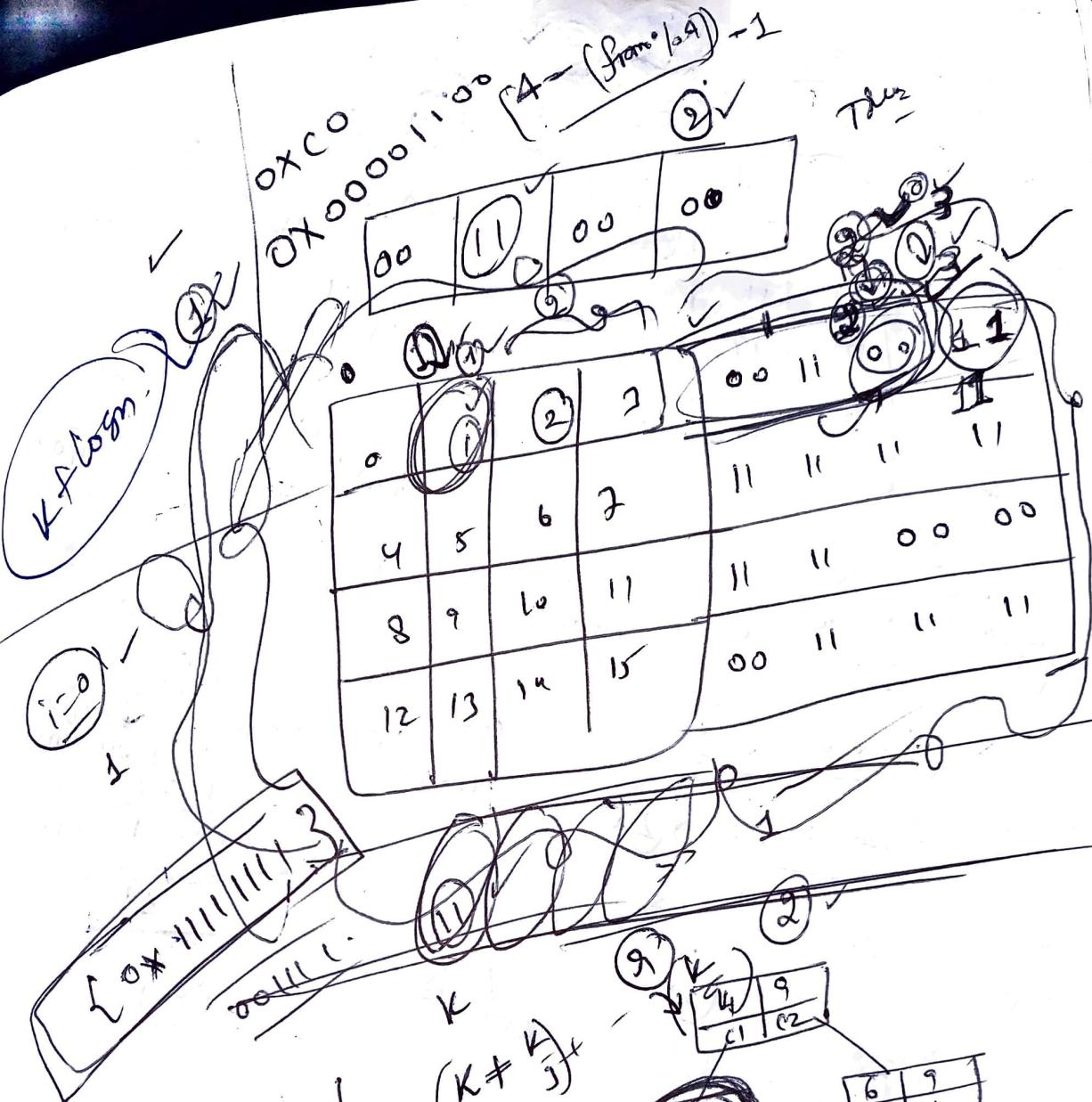
```

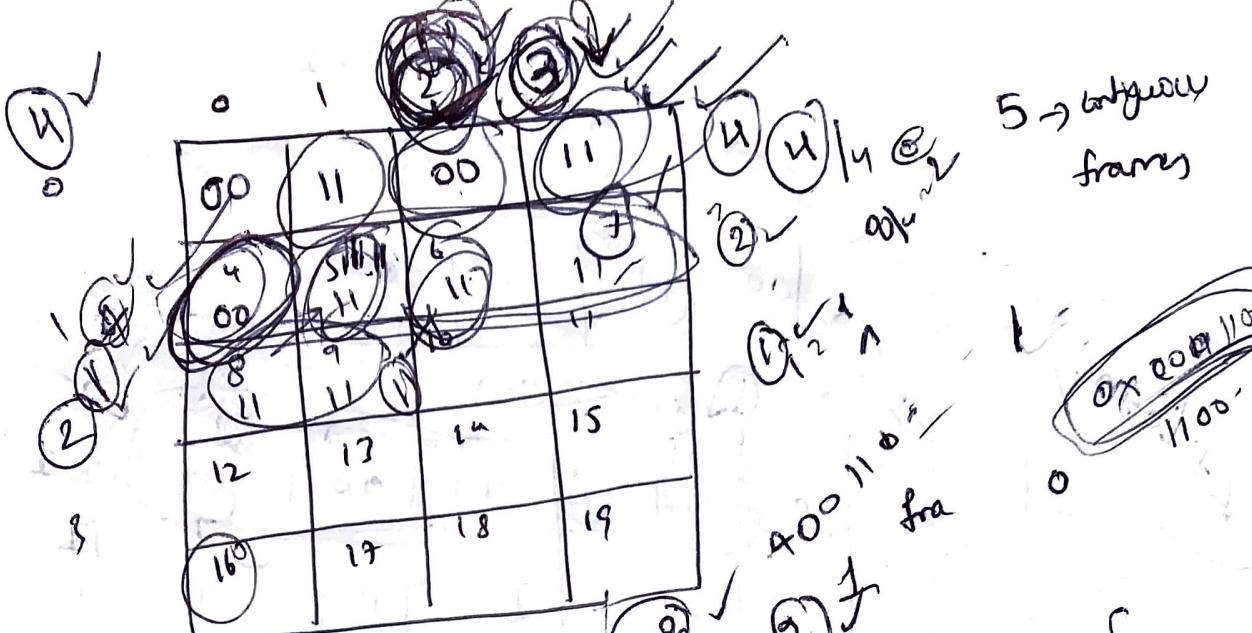
cscce410@csce410-VirtualBox: ~/Documents/MP2_Sources/M...
000000000001[ ] lt_dhandle is 0x5594194a73d0
000000000001[PLUGIN] loaded plugin libbbx_biosdev.so
000000000001[ ] lt_dhandle is 0x5594194abbd0
000000000001[PLUGIN] loaded plugin libbbx_speaker.so
000000000001[ ] lt_dhandle is 0x5594194aa950
000000000001[PLUGIN] loaded plugin libbbx_exttpuirq.so
000000000001[ ] lt_dhandle is 0x5594194ab120
000000000001[PLUGIN] loaded plugin libbbx_parallel.so
000000000001[ ] lt_dhandle is 0x5594194acd80
000000000001[PLUGIN] loaded plugin libbbx_serial.so
000000000001[ ] lt_dhandle is 0x5594194b1180
000000000001[PLUGIN] loaded plugin libbbx_gamreport.so
000000000001[ ] lt_dhandle is 0x5594194b19b0
000000000001[PLUGIN] loaded plugin libbbx_iodebug.so
000000000001[ ] reading configuration from bochsrc.bxrc
000000000001[ ] lt_dhandle is 0x5594194b2460
000000000001[PLUGIN] loaded plugin libbbx_x.so
000000000001[ ] installing x module as the Bochs GUI
000000000001[ ] using log file bochsrc.txt
Next at t=0
(0) [0x0000fffff0] f000:fff0 (unk. ctxt): jmpf 0xf000:e05b ; ea5be000
f0
<bochs:1> c

```

The log concludes with a message: "Testing is DONE. We will do nothing forever. Feel free to turn off the machine now."

The following document contains all the work done before and after drafting the code using notepad++.





5 → temporary frames

frame no = 0

Count = 1

0xC02
mask

~~0x00000000~~
~~0x00000000~~
~~0x00000000~~
~~0x00000000~~

frame no = 1
frame no = 2
frame no = 3
frame no = 4
frame no = 5
frame no = 6
frame no = 7
frame no = 8
frame no = 9
frame no = 10
frame no = 11
frame no = 12
frame no = 13
frame no = 14
frame no = 15
frame no = 16
frame no = 17
frame no = 18
frame no = 19

Count = 1

frame no = 1
frame no = 2
frame no = 3
frame no = 4
frame no = 5
frame no = 6
frame no = 7
frame no = 8
frame no = 9
frame no = 10
frame no = 11
frame no = 12
frame no = 13
frame no = 14
frame no = 15
frame no = 16
frame no = 17
frame no = 18
frame no = 19

Count = 2

0xA11111
0x00001111

0x0000001111
0x0000001111
0x0000001111
0x0000001111

Simple frame prn

11 may be at

0011 0000 00

11 00 00 00

00 00-00 11

00 00 11 00

0x30

0x60

0x03

0x0C

0x00

11 means many be

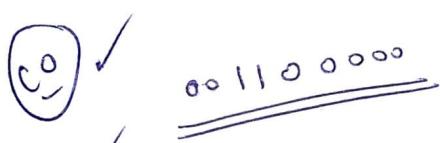
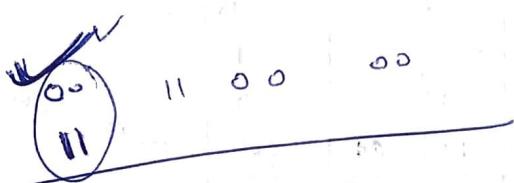
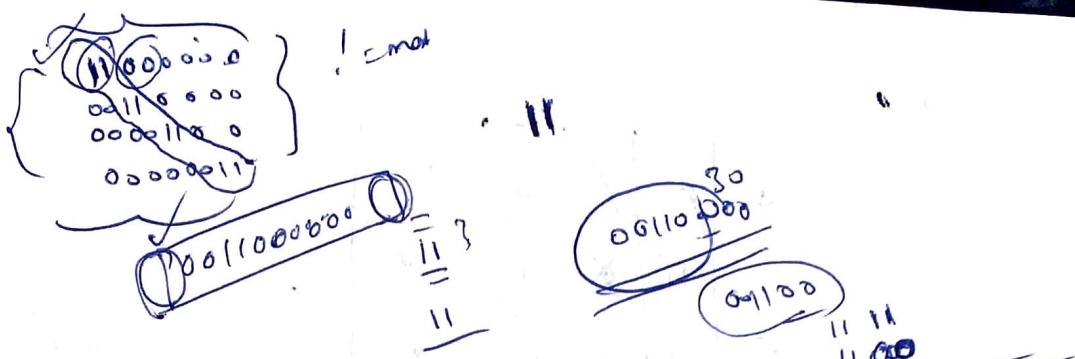
12 ✓

13

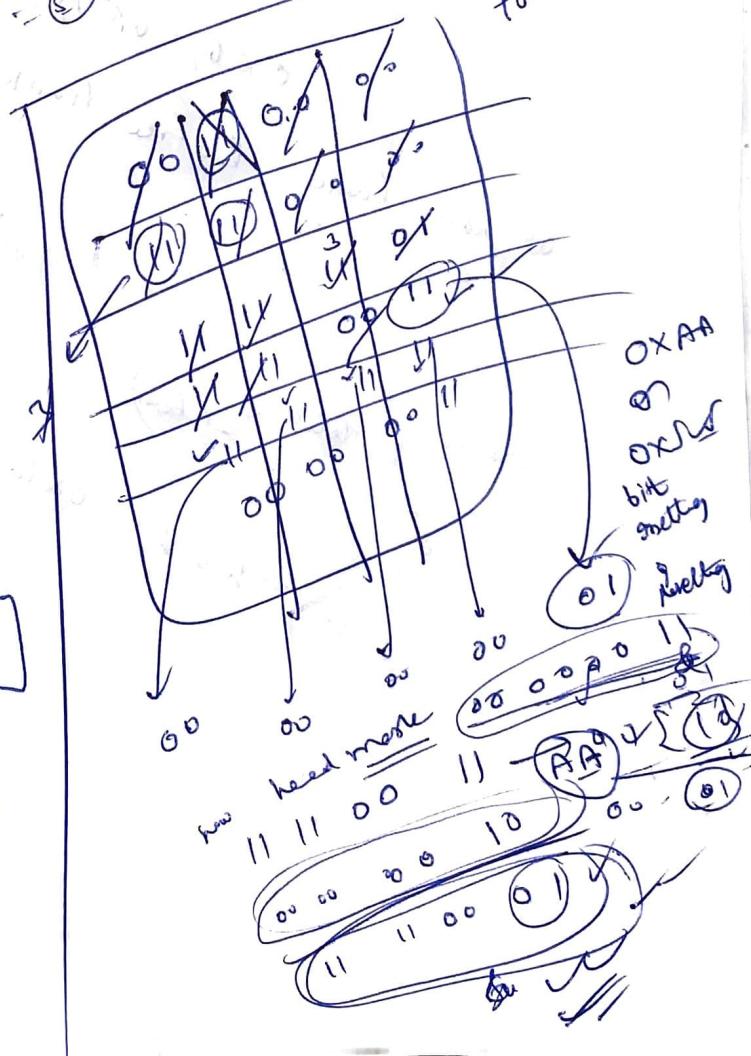
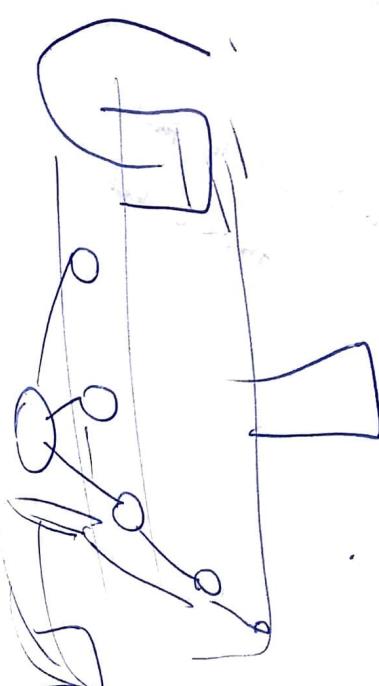
get frames

0	1	2	3
00	11	00	00
4	5	6	7
11	00	00	11
8	9	3	11
11	00	11	11
12	11	11	11
11	11	11	11

Frame	Count
0	0
1	1
2	0
3	0
4	1
5	0
6	0
7	0
8	1
9	0
10	1
11	0
12	1
13	0
14	1
15	0
16	1
17	0
18	1
19	0
20	1
21	0
22	1
23	0
24	0
25	0
26	0
27	0
28	0
29	0
30	0
31	0
32	0
33	0
34	0
35	0
36	0
37	0
38	0
39	0
40	0
41	0
42	0
43	0
44	0
45	0
46	0
47	0
48	0
49	0
50	0
51	0
52	0
53	0
54	0
55	0
56	0
57	0
58	0
59	0
60	0
61	0
62	0
63	0
64	0
65	0
66	0
67	0
68	0
69	0
70	0
71	0
72	0
73	0
74	0
75	0
76	0
77	0
78	0
79	0
80	0
81	0
82	0
83	0
84	0
85	0
86	0
87	0
88	0
89	0
90	0
91	0
92	0
93	0
94	0
95	0
96	0
97	0
98	0
99	0
100	0
101	0
102	0
103	0
104	0
105	0
106	0
107	0
108	0
109	0
110	0
111	0
112	0
113	0
114	0
115	0
116	0
117	0
118	0
119	0
120	0
121	0
122	0
123	0
124	0
125	0
126	0
127	0
128	0
129	0
130	0
131	0
132	0
133	0
134	0
135	0
136	0
137	0
138	0
139	0
140	0
141	0
142	0
143	0
144	0
145	0
146	0
147	0
148	0
149	0
150	0
151	0
152	0
153	0
154	0
155	0
156	0
157	0
158	0
159	0
160	0
161	0
162	0
163	0
164	0
165	0
166	0
167	0
168	0
169	0
170	0
171	0
172	0
173	0
174	0
175	0
176	0
177	0
178	0
179	0
180	0
181	0
182	0
183	0
184	0
185	0
186	0
187	0
188	0
189	0
190	0
191	0
192	0
193	0
194	0
195	0
196	0
197	0
198	0
199	0
200	0
201	0
202	0
203	0
204	0
205	0
206	0
207	0
208	0
209	0
210	0
211	0
212	0
213	0
214	0
215	0
216	0
217	0
218	0
219	0
220	0
221	0
222	0
223	0
224	0
225	0
226	0
227	0
228	0
229	0
230	0
231	0
232	0
233	0
234	0
235	0
236	0
237	0
238	0
239	0
240	0
241	0
242	0
243	0
244	0
245	0
246	0
247	0
248	0
249	0
250	0
251	0
252	0
253	0
254	0
255	0
256	0
257	0
258	0
259	0
260	0
261	0
262	0
263	0
264	0
265	0
266	0
267	0
268	0
269	0
270	0
271	0
272	0
273	0
274	0
275	0
276	0
277	0
278	0
279	0
280	0
281	0
282	0
283	0
284	0
285	0
286	0
287	0
288	0
289	0
290	0
291	0
292	0
293	0
294	0
295	0
296	0
297	0
298	0
299	0
300	0
301	0
302	0
303	0
304	0
305	0
306	0
307	0
308	0
309	0
310	0
311	0
312	0
313	0
314	0
315	0
316	0
317	0
318	0
319	0
320	0
321	0
322	0
323	0
324	0
325	0
326	0
327	0
328	0
329	0
330	0
331	0
332	0
333	0
334	0
335	0
336	0
337	0
338	0
339	0
340	0
341	0
342	0
343	0
344	0
345	0
346	0
347	0
348	0
349	0
350	0
351	0
352	0
353	0
354	0
355	0
356	0
357	0
358	0
359	0
360	0
361	0
362	0
363	0
364	0
365	0
366	0
367	0
368	0
369	0
370	0
371	0
372	0
373	0
374	0
375	0
376	0
377	0
378	0
379	0
380	0
381	0
382	0
383	0
384	0
385	0
386	0
387	0
388	0
389	0
390	0
391	0
392	0
393	0
394	0
395	0
396	0
397	0
398	0
399	0
400	0
401	0
402	0
403	0
404	0
405	0
406	0
407	0
408	0
409	0
410	0
411	0
412	0
413	0
414	0
415	0
416	0
417	0
418	0
419	0
420	0
421	0
422	0
423	0
424	0
425	0
426	0
427	0
428	0
429	0
430	0
431	0
432	0
433	0
434	0
435	0
436	0
437	0
438	0
439	0
440	0
441	0
442	0
443	0
444	0
445	0
446	0
447	0
448	0
449	0
450	0
451	0
452	0
453	0
454	0
455	0
456	0
457	0
458	0
459	0
460	0
461	0
462	0
463	0
464	0
465	0
466	0
467	0
468	0
469	0
470	0
471	0
472	0
473	0
474	0
475	0
476	0
477	0
478	0
479	0
480	0
481	0
482	0
483	0
484	0
485	0
486	0
487	0
488	0
489	0
490	0
491	0
492	0
493	0
494	0
495	0
496	0
497	0
498	0
499	0
500	0
501	0
502	0
503	0
504	0
505	0
506	0
507	0
508	0
509	0
510	0
511	0
512	0
513	0
514	0
515	0
516	0
517	0
518	0
519	0
520	0
521	0
522	0
523	0
524	0
525	0
526	0
527	0
528	0
529	0
530	0
531	0
532	0
533	0
534	0
535	0
536	0
537	0
538	0
539	0
540	0
541	0
542	0
543	0
544	0
545	0
546	0
547	0
548	0
549	0
550	0
551	0
552	0
553	0
554	0
555	0
556	0
557	0
558	0
559	0
560	0
561	0
562	0
563	0
564	0
565	0
566	0
567	0
568	0
569	0
570	0
571	0
572	0
573	0
574	0
575	0
576	0
577	0
578	0
579	0
580	0
581	0
582	0
583	0
584	0
585	0
586	0
587	0
588	0
589	0
590	0
591	0
592	0
593	0
594	0
595	0
596	0
597	0
598	0
599	0
600	0
601	0
602	0
603	0
604	0
605	0
606	0
607	0
608	0
609	0
610	0
611	0
612	0
613	0
614	0
615	0
616	0
617	0
618	0
619	0
620	0
621	0
622	0
623	0
624	0
625	0
626	0
627	0
628	0
629	0
630	0
631	0
632	0
633	0
634	0
635	0
636	0
637	0
638	0
639	0
640	0
641	0
642	0
643	0
644	0
645	0
646	0
647	0
648	0
649	0
650	0
651	0
652	0
653	0
654	0
655	0
656	0
657	0
658	0
659	0
660	0
661	0
662	0
663	0
664	0
665	0
666	0
667	0
668	0
669	0
670	0
671	0
672	0
673	0
674	0
675	0
676	0
677	0
678	0
679	0



AA



Finally

0	1	2	3
4	00	00	00
5	00	00	00
6	00	00	00
7	00	00	00
8	00	00	00
9	00	00	00
10	00	00	00
11	00	00	00
12	00	00	00
13	00	00	00
14	00	00	00
15	00	00	00
16	00	00	00
17	00	00	00
18	00	00	00
19	00	00	00
20	00	00	00
21	00	00	00
22	00	00	00
23	00	00	00

final index

So finally

head pos
0x1000000

while checking for

marks using

0x00, 0x04

0x30, 0x03

framecat index = 51 position
of free

we have found

offset = 0

while (count < nframes)

0x08 shift to that
frame offset

{ mark with
page }

get-frames

	0	1	2	3
0	00	11	00	00
1	11	11	00	00
2	10	10	10	11
3	11	11	11	11
4	10	11	10	11

Ngr

Nat Next

4

Port Ward

11

Mart

14