Procedure:

- 1) The handout is clearly studied and
- 2) The first 4MB is directly mapped
- 3) Page tables are setup correctly
- 4) Paging is enabled
- 5) The CRO, CR2, CR3 registers are properly utilised
- 6) The page table entries are correctly masked for address space beyond 4MB
- 7) The testing is done in new development environment using the following commands
- 8) make, ./copykernel.sh, bochs -f bochsrc.bxrc
- 9) The following photo shows the result of the output.
- 10) The detailed paper work is attached at the end of this document.
- 11) The github link is https://github.tamu.edu/kausht14/OSMP3

Result:

```
Installing handler in IDT position 43
Installing handler in IDT position 44
Installing handler in IDT position 45
Installing handler in IDT position 46
Installing handler in IDT position 47
Installed exception handler at ISR <0>
Installed interrupt handler at IRQ <0>
Installed interrupt handler at IRQ <1>
Frame Pool initialized
Frame Pool initialized
Installed exception handler at ISR <14>
Initialized Paging System
Constructed Page Table object
Loaded page table
Enabled paging
WE TURNED ON PAGING!
If we see this message, the page tables have been
set up mostly correctly.
Hello World!
EXCEPTION DISPATCHER: exc no = <14>
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
```

```
Landled page fault

EXCEPTION DISPATCHER: exc_no = <14>
handled page fault

DONE WRITING TO MEMORY. Now testing...

TEST PASSED

YOU CAN SAFELY TURN OFF THE MACHINE NOW.

One second has passed

One second has passed
```

	1 2 2 1
	umb directly mapped
(1)	Memory - 4MB - directly mapped
U	a 1-ite allocate in the page
	Memory - 4MB - solvente in the page table
	· · · · · · · · · · · · · · · · · · ·
	Beyond ums read implementation
(2)	
	DATI PR
(3)	only one process
	First 4mB of each address space
	First LMB of each was to LMB of physical montery
	maps
	Create a page take orgical to access the
_	Create a page take orgent to access memory frames directory -> page take pager > memory frames
(h)	Line pages memory
	BOY WE TO
	the stark
	O Store the page table orgent on the stark
(5)	O Store the T
0	may take (befre to)
	O Store the page table object out Store the page table befre paging endred Sourcete page table befre paging endred
	to prevent page fault
	to present
	la using current variable
	Then load the page table by using current variable
6	Then (1884)
	to store court take
	once toaded enable paging dry using the
	Once toaded enable paging dry asing CRO Registed paging but asm & paging-bouty
(7)	paging bu asm & paging - w
	(CRO Register)
	C 1 later
6	hardling faults later is available
8	haranery is awaitable
	get-traver & release Frames is available
(9)	get-"
	and the first the state of the

(18) Kernel poor - 2mp & umb & is in direct mapped
mandey
11) process-mens-post above AMB & freely mapped
-> define init paging & store into private
Variables Natures probled
-> Once init done, Kernel sels up first page take
(oyeet)
→ Give a fram post to directory
we need to configure port first
-> After init, pools mit, boar the page take
do Rea CP3 - & page take - directory
one sure dung content switch the
Late Dogs Vs lacoled
-> Switch to paging way enoble possing () by
Nothing CRO Register
-> Only ene do hardle-fault() after introdistation
of pege table is done.
00.10

Below umB -> deally mapped whereve above AMA, page fault occurs & page - fault handler should O get-free-frame from free-frame prot Should allocate this to process Update the proge entry } if multiple references Oper again getres the insmeter Store the page directory, pagetally pages & management into for the process trave pool in -) Management ma for kend is however stored in Kend por itself because dhadly mapped (0-4MB) Don't forget cro, cry (f) CRO - page endling OPS - page directory table inglisher Structure of PT entry can be used for setting use fairty tolls it 5 4-3 210 31 - . . 12 11 . . . 9 8 . . 7 6 Page frame Avail Resurce (D) (A) U/s- User of Supervisa level R/W - Reed or write Project - Use bit

A page fault triggors exection 14" -> pushes a wood with the exception crun code on to the stack -> REGIS argument tells about the exception ___ Lower 2 bits of word pushed onto the Stark Vilve 2 1 0

Never 2 1 0

Kennel geet page not present of ever code tolde to charle when white protection fault 32 bit address of the address that caused the x page-fault -> Star in CR2 / and head weig CR_2 > geod-002(). 1) (Set the your address (Valid bit to high) made the invalid bit initially for above the then page - faults ocan, puts in page fall t the sets to valid, mark the progress present natura from exception

-> not prosent uma for Phow wit 0 perong to page direct Page talle inder = 3kg 10 bits 1024 12 bits withouty & we need boxt only this reded shill by OXO3 FF