

KERNEL KREW SPRINT 2 REPORT

GITHUB LINK: https://github.com/reineB5/KernelKrew_xv6

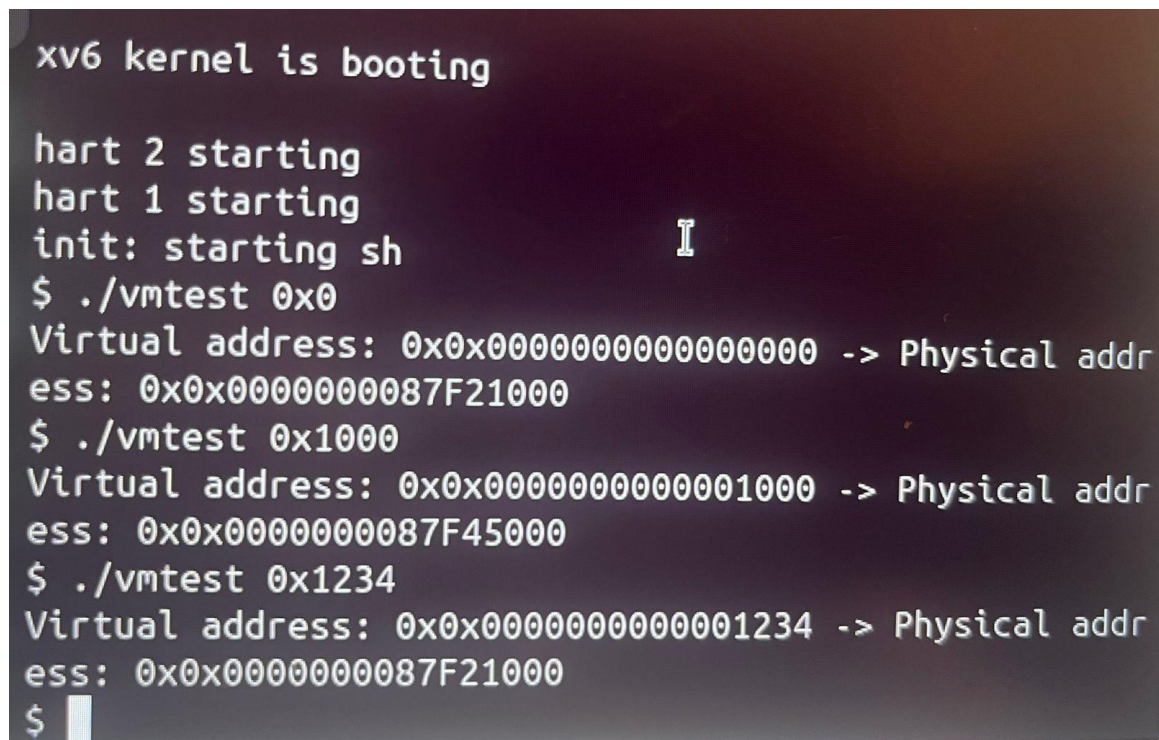
New Feature:

- **vtop syscall:** A new system call that translates a given virtual address to its corresponding physical address in xv6.

Testing:

We tested the syscall using a custom user program vmtest.

Example commands inside xv6:



```
xv6 kernel is booting

hart 2 starting
hart 1 starting
init: starting sh
$ ./vmtest 0x0
Virtual address: 0x0x0000000000000000 -> Physical address: 0x0x0000000087F21000
$ ./vmtest 0x1000
Virtual address: 0x0x00000000000001000 -> Physical address: 0x0x0000000087F45000
$ ./vmtest 0x1234
Virtual address: 0x0x00000000000001234 -> Physical address: 0x0x0000000087F21000
$
```

Modifications:

kernel/syscall.c

- Declared the system call handler: `extern uint64 sys_vtop(void);`
- Mapped the system call number to the handler: `[SYS_vtop] sys_vtop,`

user/user.h

-Declared the user-level interface for the new syscall: int vtop(void* vaddr);

user/vtopstub.c (*New file*)

-Implemented the user-side syscall wrapper:

kernel/sysproc.c

- Added the function uint64 sys_vtop(void) to fetch the user-supplied virtual address and call walkaddr() to obtain the physical address.

kernel/syscall.c

- Defined a syscall number SYS_vtop in the enum.
- Added an entry for SYS_vtop in the syscalls[] function pointer array.

user/usys.S

- **Declared the syscall stub:**

.global vtop

vtop:

li a7, SYS_vtop

ecall

ret

user/vmtest.c

- Created a user program vmtest that accepts a virtual address as input, invokes vtop, and prints the result.

Makefile

- Added the following entry under UPROGS:
\$U/_vmtest\

kernel/syscall.h

- Defined:

```
#define SYS_vtop <number>
```

(*The syscall number matches the one used in syscall.c and usys.S.*)

user/vmtest.h

To avoid build errors caused by the default user.h and types.h in xv6, we created a minimal custom header file (user/vmtest.h) for the vmtest program. This header defines the necessary types such as uint64 and declares only the essential function prototypes (vtop, printf, and exit). This approach simplified compilation and prevented implicit declaration errors by explicitly specifying the interfaces used in the user test program