

```
$ uv2p 0
invalid physical address: 0x80000000
$ uv2p 80000000
physical address: 0x0
$ uv2p 8a000000
physical address: 0xA000000
$ uv2p 8dffffff
physical address: 0xDFFFFFFF
$ uv2p 8e000000
invalid physical address: 0xE000000
$ uv2p a0000000
invalid physical address: 0x20000000
$ uv2p ffffffff
invalid physical address: 0x7FFFFFFF
$
```

Figure 1: Various Test Cases

As physical memory uses 0x00000000 – 0xE0000000 to represent virtual addresses, test cases should encompass valid virtual addresses from 0x00000000 to 0xFFFFFFFF (all available spaces in total memory).

Test 0x0 checks first virtual user space and demonstrates truncation of hex address (code specific to allow easier addressing). 0x0 represent first user text in virtual space, which exists outside of physical memory or below KERNBASE (0x80000000) therefore is invalid, which the code demonstrates.

Test 0x80000000 checks the first valid virtual address, in other words where KERNBASE is defined, which returns a valid address and the first physical memory.

Test 0x8A000000 demonstrates an address between KERNBASE and (KERNBASE + PHYSTOP), or an address between the bounds. This returns a valid address between 0 and PHYSTOP.

Test 0x8DFFFFFF checks the last valid virtual byte right before KERNBASE + PHYSTOP.

Test 0x8E000000 checks PHYSTOP, or the first upper bound invalid address.

Test 0xA000000 checks further above.

Test 0xFFFFFFFF checks the last possible byte.