

Test Cases for uv2p()

(1) Test1: create an array of 5 elements 1, 2, 3, 4, 5. Then we use a loop to iterate the array and get the physical address of each element in the array.

```
int a[5]={1,2,3,4,5};  
if(uv2p((char*)(a+i))<0)
```

(2) Test2: create a character "a". Then we use uv2p to translate the the character's virtual address to physical address.

```
char c = 'a';  
if(uv2p(&c)<0)
```

(3) Test3: let the system call to translate a virtual address that is out of bound, expecting the system is returning false;

Test Result:

Using gdb to get the content of physical addresses, the returned elements were the 5 integers and the character "a";

This ensure the correctness of the system call to translate virtual address to physical address. Please see the picture below of gdb result.

```

xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ myProgram

physical address is: dedffb8, a[0] virtual address is: 2FB8
physical address is: dedffbc, a[1] virtual address is: 2FBC
physical address is: dedffc0, a[2] virtual address is: 2FC0
physical address is: dedffc4, a[3] virtual address is: 2FC4
physical address is: dedffc8, a[4] virtual address is: 2FC8
physical address is: dedffcf, c virtual address is: 2FCF QEMU 1.0 monitor - type 'help' for more information
(qemu) xp/x 0xdedffb8
00000000dedffb8: 0x00000001
(qemu) xp/x 0xdedffbc
00000000dedffbc: 0x00000002
(qemu) xp/x 0xdedffc0
00000000dedffc0: 0x00000003
(qemu) xp/x 0xdedffc4
00000000dedffc4: 0x00000004
(qemu) xp/x 0xdedffc8
00000000dedffc8: 0x00000005
(qemu) xp/c 0xdedffcf
00000000dedffcf: 'a' '\x00' '\x00' '\x00'
(qemu)

```

Failed result

```

not allocated virtual address is : B000 input virtual address not exist in page
table

```

Explanation

The data is of good quality because

- 1) our test cases allocate memory for different data types(int, char);
- 2) the growth of memory address will changed because of the size of data types;
- 3) handling exception(address out of boundary) was included in the code by returning "input virtual address not exist in page table"