

Below are the steps to add a new hypercall call to a hypervisor (xen):

- 1) Add new hypercall in **bold** to /usr/src/linux/include/xen/interface/xen.h:

```
#define __HYPERVISOR_arch_7          55
#define __HYPERVISOR_goodbyeworld 56
```

- 2) Add it to entry.S

a) Add to hypercall table:

```
.quad goodbyeworld      /* 56 */
```

b) Add it to hypercall_args_table

```
.byte 0 /* goodbyeworld */ /* 56 */
```

- 3) Add declaration to /usr/src/linux/arch/x86/include/asm/xen/hypercall.h:

```
long goodbyeworld(void);
```

- 4) Add implementation in xen/common/domctl.c:

```
long goodbyeworld(void)
{
    printk("goodbye world!");
}
```

- 5) Add function to call this new hypercall in xen/tools/xc_domain.c:

```
int hypercall_test(int handle)
{
    int rc;
    int arg=0;

    DECLARE_HYPERCALL;
    hypercall.op = __HYPERVISOR_goodbyeworld;
    rc = do_xen_hypercall(handle, &hypercall);
    hypercall.arg[0] = 0;
    hypercall.arg[1] = (unsigned long)&arg;
    return rc;
}
```

- 6) Write userlevel program (*vi test_hypercall.c*) to call function hypercall_test and invoke new hypercall

```
#include <xenctrl.h>
#include <stdio.h>
int main()
{
    printf("Attempt to invoke the hypercall: __HYPERVISOR_goodbyeworld\n");
    int handle, rc;

    /* Acquire Hypervisor Interface Handle.
    This handle goes as the first argument for the function do_xen_hypercall()
    */

    handle = xc_interface_open();
    printf ("Acquired handle to Xen Hypervisor:%d\n",handle);

    rc = hypercall_test(handle);
    printf ("Hypercall Details: %d\n", rc);

    xc_interface_close(handle);
    printf ("Hypervisor handle closed\n");

    return 0;
}
```

- 7) Open a new terminal and monitor /var/log/syslog

```
tail -f /var/log/syslog
```

- 8) Compile and run user level program

```
gcc test_hypercall.c -o test_hypercall
./test_hypercall
```