

Invoking the OOM Killer (Lab)

coursera.org/learn/linux-for-developers/supplement/miVk7/invoking-the-oom-killer-lab

Exercise

When the Linux kernel gets under extreme memory pressure, it invokes the dreaded OOM (Out Of Memory) Killer. This tries to select the "best" process to kill to help the system recover gracefully.

We are going to force the system to run short on memory and watch what happens. The first thing to do is to open up a terminal window, and in it, type:

```
$ sudo tail -f /var/log/messages
```

in order to watch kernel messages as they appear.

An even better way to look is furnished by:

```
$ dmesg -w
```

as it does not show non-kernel messages.

This exercise will be easier to perform if we turn off all swap first with the command:

```
$ sudo /sbin/swapoff -a
```

Make sure you turn it back on later with:

```
$ sudo /sbin/swapon -a
```

Now, we are going to put the system under increasing memory pressure. You are welcome to find your own way of doing it, but we also supply a program for consuming the memory:

```
/* simple program to defragment memory, J. Cooperstein 2/04
```

```
@*/
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <unistd.h>
```

```

#include <string.h>

#define MB (1024*1024)

int main(int argc, char **argv)
{
    int j, m;

    char *c;

    if (argc < 2) {
        fprintf(stderr,
            "You must give the memory in MB to waste, aborting\n");
        exit(EXIT_FAILURE);
    }

    m = atoi(argv[1]);

    for (j = 0; j < m; j++) {
        /* yes we know this is a memory leak, no free, that's the idea! */
        c = malloc(MB);
        memset(c, j, MB);
        printf("%5d", j);
        fflush(stdout);
    }

    printf("All memory allocated, pausing 5 seconds\n");

    sleep(5);

    printf("Quitting and releasing memory\n");

    exit(EXIT_SUCCESS);
}

```

It takes as an argument how many MB to consume. Keep running it, gradually increasing the amount of memory requested, until your system runs out of memory.

Note: You should be able to compile the program by just doing **gcc -o lab_wastemem lab_wastemem.c**, or get the binary from your instructor.

You should see the OOM (Out of Memory) killer swoop in and try to kill processes in a struggle to stay alive. Who gets clobbered first?

Solution

You can find one possible solution below:

[lab_waste.sh](#)

[lab_wastemem.c](#)

,