

## Lab #5: read

### Overview

The purpose of this lab is for you to gain experience with the costs of system calls and to explore some of the limits imposed on resources provided by the system.

### Part 1: read

Write a simple C program that opens a file (`/usr/lib/locale/locale-archive` on the unix? machines is an ok choice due to its size (though it is rather small by today's standards) and reads the contents of that file using the `read` system call. Write this code in a manner that allows you to change the number of bytes read (use a `#define` for the array size and argument to `read`).

Run this program with sizes **1, 2, 16, 32, 64, 128, 256, 512, 1024, 2048** and record the running time with the `time` program. Plot the results to see if you can determine any characteristics about the system.

Your program should not print anything.

### Part 2: fread

Do the above with the `fread` library function in place of the `read` system call (and, of course, `fopen` instead of `open`).

### Part 3: Limits

Write a program that includes `unistd.h` and `limits.h`. This program should use `sysconf` (see the man page) to print out the values for various system limits. At a minimum, you should print the values for `_SC_CHILD_MAX`, `_SC_OPEN_MAX`, and `_SC_PAGE_SIZE`.

Run this program on `unix3`, `unix13`, and the local lab machine.

### Part 4: Shell Limits

Many command shells provide a means to restrict resources for those programs executed by the shell.

If you are using `bash`, type `ulimit -a`. If you are using `tcsh`, type `limit`. You should see a list of the different limits on resources. You can change these with the specified switches for `ulimit` or with a unique prefix for the resource for `limit`.

Change the virtual memory limit to a smaller value (`-v` for `ulimit` (you will likely want `-S`) as well to set the soft limit instead of the hard limit) and `vm` for `limit`). See what happens if you set it too low.

#### malloc

Write a small program that allocates ten megabytes of memory. Set the virtual memory limit high enough for this program to run, but low enough, to cause `malloc` to fail.

### Demonstration

Demonstrate each of the above parts to receive credit for this lab.