Due: Tuesday, Mar 16, 2009. 11:59pm (no extensions).

What to submit: Upload a ASCII text files with your answers.

As some answers are specific to our current environment, you should do this exercise on our rlogin cluster or the lab machines.

1. Observing System Calls

Processes use system calls to obtain services from the kernel. The program ‘strace’ is a utility that can display which system calls a program is currently executing (if any). Familiarize yourselves with the ‘strace’ command (use ‘man strace’, for instance). In particular, note the switches ‘-f’, ‘-e’, and ‘-p’.

1. Hello World. Consider the following program:

```c
#include <stdio.h>
int
main()
{
    printf("Hello, World\n");
}
```

Build the program with `gcc -static -o hello.static hello.c` which will link the C library statically. Run the program with `strace`.

Which system call performs the output of the “Hello, World” message? Copy the output of strace.

2. The Shell. The ‘-c’ switch asks the bash program to interpret the following argument as a series of commands and to execute them. Run the following commands:

(a) `strace -f -e process bash -c "/bin/echo 1; exit;"

Create an ASCII sketch showing which process executes which process-related system call.

(b) Now run

`strace -f -e process,pipe,dup2 bash -c "ls | wc"

Create an ASCII sketch showing which system call (process-related, dup2, or pipe) is executed by which process.

2. System Call Or Not

For each of the following functions, determine if they are system calls or not. If they aren’t system calls, list the system call or system calls that is invoked to implement them.
Provide the concrete Linux names. Note that Linux, for historical reasons, has many of its own names and variants for what are in other Unixes actual system calls. Furthermore, these names may sometimes change from Linux version to Linux version. Provide the answers for our current lab machines.

Write small test programs to obtain the answers, as the information you’ll find online or sometimes even in man pages may be out of date.

1. fork()
2. exit()
3. sigprocmask()
4. system() (Use strace -e process)
5. wait()
6. sleep()

3. Why Is This Program Stuck?

A common diagnostic task is to look at a process that is ‘stuck’ - it has stopped responding to input and does not produce any output.

`cs3214/public_html/spring2010/exercises/sys1/` contains 7 programs, stuck1 through stuck7. All programs, when run from the command line, appear to get ‘stuck’ immediately. Find out what each of these programs appears to be doing. Collect as much information as you can via ps, /proc/pid, strace.