

Operating Systems

Assignment 1

Krunal Shah (2015EE10476)

1 System call tracing

1.1 Trace printing

The printing of the system calls is implemented inside the function `syscall` from where the system calls are called using the array of function pointers `syscalls[]` in the file `syscall.c` and the following variables were added to the file `syscall.c`

1. `int num_calls[24]`: keeps track of the number of times the system calls are called
2. `const char* call_name[]`: stores the string to be printed(name of the system call) when a system call is called

1.2 `sys_toggle()` System Call

The system call toggle was created as follows:

1. An integer variable `toggle` was added to the file `syscall.c` which maintains whether the printing of the system calls log is enabled or disabled. The variable is initialised with 1.
2. The function `sys_toggle` is implemented in the file `syscall.c` which toggles the value of the variable `toggle`.
3. The if condition is added before the printing of the system calls trace in the function `syscall` implemented in the file `syscall.c`
4. `#define SYS_toggle 22` is added to the file `syscall.h`
5. `[SYS_toggle] sys_toggle` is added to the array of function pointers `syscalls`
6. `SYSCALL(toggle)` is added to the file `usys.S`
7. `int toggle(void)` is added to the file `user.h`
8. `sys_toggle` entries are added to `num_calls` and `call_name` arrays for trace printing of the system call

2 `sys_add()` System Call

The system call `add` was created as follows:

1. The function `sys_add` is implemented in the file `proc.c` and the function obtains the values of the arguments passed to the invocation of the system call by using the function `argint`.
2. `#define SYS_add 23` is added to the file `syscall.h`
3. `extern int sys_add(void)` is added to the file `syscall.c`
4. `[SYS_add] sys_add` is added to the array of function pointers `syscalls`
5. `SYSCALL(add)` is added to the file `usys.S`
6. `int toggle(int, int)` is added to the file `user.h`
7. `sys_add` entries are added to `num_calls` and `call_name` arrays for trace printing of the system call

3 `sys_ps()` System Call

The system call `ps` was created as follows:

1. The function `sys_ps` is implemented in the file `proc.c` and the function iterates through the processes in the variable `ptable` and then prints the process information for the processes for which `(pid > 0)` AND `(state in {RUNNABLE, RUNNING, SLEEPING})`
2. `#define SYS_ps 24` is added to the file `syscall.h`
3. `extern int sys_ps(void)` is added to the file `syscall.c`
4. `[SYS_ps] sys_ps` is added to the array of function pointers `syscalls`
5. `SYSCALL(ps)` is added to the file `usys.S`
6. `int ps(void)` is added to the file `user.h`
7. `sys_ps` entries are added to `num_calls` and `call_name` arrays for trace printing of the system call