- Michael Smith

Lab5

CSE 460

2/06/2018

Total Points 20

1. Message Queues

    Command Studies

    - Msgctl performs the control operation specified by cmd.
    - Msgget returns the message queue associated with the value of the key argument
    - Msgrcv receives then reads the message from the specified queue id.
    - Msgsend sends the message to the queue

Msg1.cpp

```c
//msg1.cpp
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <unistd.h>

#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#define MAX_TEXT 512

struct my_msg_st {
    long int my_msg_type;
    char some_text[BUFSIZ];
};

int main()
{
    int running = 1;
    int msgid, msgid1;
    struct my_msg_st some_data;
    long int msg_to_receive = 0;
    char buffer[BUFSIZ];

/* First, we set up the message queue. */

    msgid = msgget((key_t)1234, 0666 | IPC_CREAT);

    msgid1 = msgget((key_t)1234, 0666 | IPC_CREAT);     //created a second msgid

    if (msgid == -1) {
        fprintf(stderr, "msgget failed with error: %d\n", errno);
        exit(EXIT_FAILURE);
    }
/* Then the messages are retrieved from the queue, until an end message is encountered.
 Lastly, the message queue is deleted. */

    while(running) {
        if (msgrcv(msgid, (void *)&some_data, BUFSIZ,
                   msg_to_receive, 0) == -1) {
            fprintf(stderr, "msgrcv failed with error: %d\n", errno);
            exit(EXIT_FAILURE);
        }
        printf("You wrote: %s", some_data.some_text);
        if (strncmp(some_data.some_text, "end", 3) == 0) {
            running = 0;
        }
"msg1.cpp" 69L, 1688C                                                    1,1
```

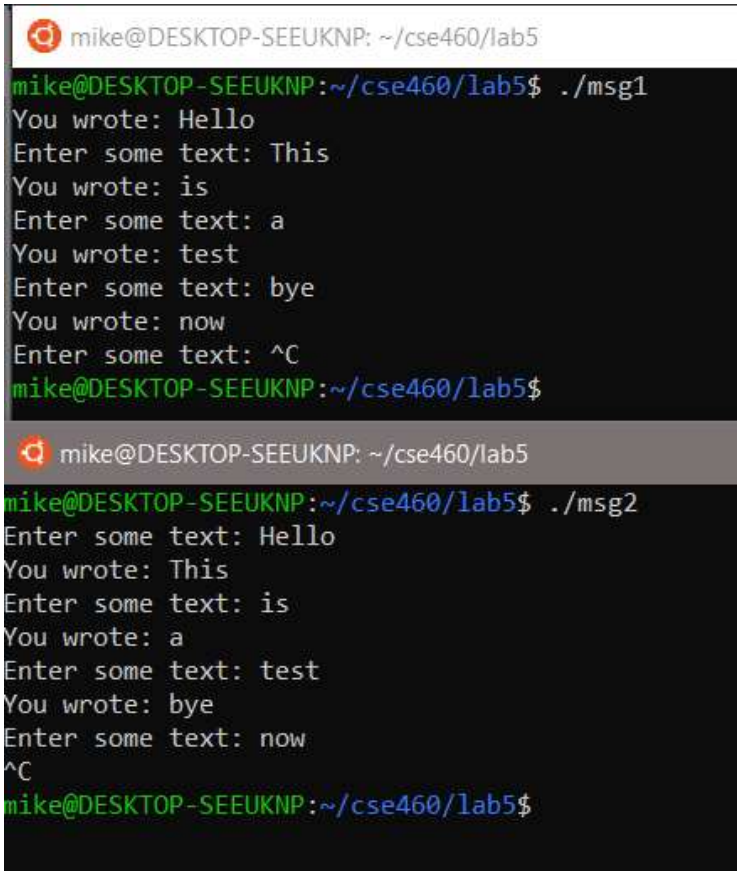
```
        printf("You wrote: %s", some_data.some_text);
        if (strncmp(some_data.some_text, "end", 3) == 0) {
            running = 0;
        }
        else{
                printf("ENter some test: ");
                fgets(buffer, BUFSIZ, stdin);
                some_data.my_msg_type = 1;
                strcpy(some_data.some_text, buffer);
                if(msgsnd(msgid, (void *)&some_data, MAX_TEXT, 0) == -1) {
                        exit(EXIT_FAILURE);
                }
                if(strncmp(buffer, "end", 3) == 0) {
                        running = 0;
                }
        }
    }

    if (msgctl(msgid, IPC_RMID, 0) == -1) {
        fprintf(stderr, "msgctl(IPC_RMID) failed\n");
        exit(EXIT_FAILURE);
    }

    exit(EXIT_SUCCESS);
}
```

Msg2.cpp

```c
// Modified message
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <unistd.h>

#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>

#define MAX_TEXT 512

struct my_msg_st {
    long int my_msg_type;
    char some_text[MAX_TEXT];
};

int main()
{
    int running = 1;
    struct my_msg_st some_data;
    int msgid, msgid1;
    char buffer[BUFSIZ];
    long int msg_to_receive = 0;

    msgid = msgget((key_t)1234, 0666 | IPC_CREAT);

    msgid1 = msgget((key_t)1234, 0666 | IPC_CREAT); // created a second msgid

    if (msgid == -1) {
        fprintf(stderr, "msgget failed with error: %d\n", errno);
        exit(EXIT_FAILURE);
    }

    while(running) {
        printf("Enter some text: ");
        fgets(buffer, BUFSIZ, stdin);
        some_data.my_msg_type = 1;
        strcpy(some_data.some_text, buffer);

        if (msgsnd(msgid, (void *)&some_data, MAX_TEXT, 0) == -1) {
            exit(EXIT_FAILURE);
        }

        else{                                                    //added this else statment
                if (msgrcv(msgid1, (void *)&some_data, BUFSIZ, msg_to_receive, 0) == -1) {
                        fprintf(stderr, "msgrcv failed with error: %d\n", errno);
"msg2.cpp" 66L, 1468C                                                                    1
```

```c
        else{                                                    //added this else statment
                if (msgrcv(msgid1, (void *)&some_data, BUFSIZ, msg_to_receive, 0) == -1) {
                        fprintf(stderr, "msgrcv failed with error: %d\n", errno);
                        exit(EXIT_FAILURE);
                }
        }

        printf("You wrote: %s", some_data.some_text);

        if (strncmp(buffer, "end", 3) == 0) {
            running = 0;
        }
    }

        if(msgctl(msgid, IPC_RMID, 0) == -1) {
                fprintf(stderr, "msgctl(IPC_RMID) failed\n");
                exit(EXIT_FAILURE);
        }

    exit(EXIT_SUCCESS);

                                                                                    66,1
```

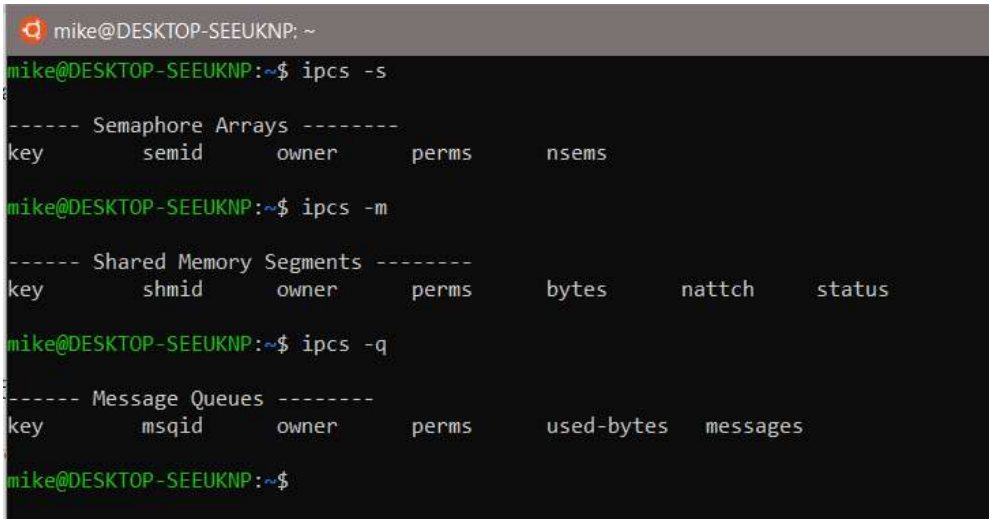
Msgs Output



2. IPC Status Commands

   Command Studies:

   - Ipcs has three modifiers. -s will identify which process is using semeaphores. And –m identifies which segment of memory is shared. Lastly –q will identify which IPC's semeaphores has messages in its queue.
   - Ipcrm – Removes the interprocess communication with the specified Sem ID



3. Study of XV6

```
mike@DESKTOP-SEEUKNP:~/cse460/temp1/xv6-public$ gdb
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
warning: File "/home/mike/cse460/temp1/xv6-public/.gdbinit" auto-loading has been declined by your `auto
-load safe-path' set to "$debugdir:$datadir/auto-load".
To enable execution of this file add
        add-auto-load-safe-path /home/mike/cse460/temp1/xv6-public/.gdbinit
line to your configuration file "/home/mike/.gdbinit".
To completely disable this security protection add
        set auto-load safe-path /
line to your configuration file "/home/mike/.gdbinit".
For more information about this security protection see the
"Auto-loading safe path" section in the GDB manual.  E.g., run from the shell:
        info "(gdb)Auto-loading safe path"
(gdb) target remote : 26000
Remote debugging using : 26000
0x0000fff0 in ?? ()
(gdb) file kernel
A program is being debugged already.
Are you sure you want to change the file? (y or n) y
Reading symbols from kernel...done.
(gdb) break swtch
Breakpoint 1 at 0x8010466b: file swtch.S, line 11.
(gdb) continue
Continuing.

Thread 1 hit Breakpoint 1, swtch () at swtch.S:11
11        movl 4(%esp), %eax
(gdb) step
12        movl 8(%esp), %edx
(gdb) step
15        pushl %ebp
(gdb) step
swtch () at swtch.S:16
16        pushl %ebx
(gdb) step
swtch () at swtch.S:17
```

```
swtch () at swtch.S:17
17        pushl %esi
(gdb) step
swtch () at swtch.S:18
18        pushl %edi
(gdb) step
swtch () at swtch.S:21
21        movl %esp, (%eax)
(gdb) step
22        movl %edx, %esp
(gdb) step
swtch () at swtch.S:25
25        popl %edi
(gdb) step
swtch () at swtch.S:26
26        popl %esi
(gdb) step
swtch () at swtch.S:27
27        popl %ebx
(gdb) step
swtch () at swtch.S:28
28        popl %ebp
(gdb) step
swtch () at swtch.S:29
29        ret
(gdb) step
forkret () at proc.c:398
398     {
(gdb) step
forkret () at proc.c:401
401       release(&ptable.lock);
(gdb) step
release (lk=0x80112d20 <ptable>) at spinlock.c:48
48      {
(gdb) step
49        if(!holding(lk))
(gdb) continue
Continuing.

Thread 1 hit Breakpoint 1, swtch () at swtch.S:11
11        movl 4(%esp), %eax
(gdb) clear
Deleted breakpoint 1
(gdb) break exec
Breakpoint 2 at 0x801009f0: file exec.c, line 12.
(gdb) continue
Continuing.
[Switching to Thread 2]
```

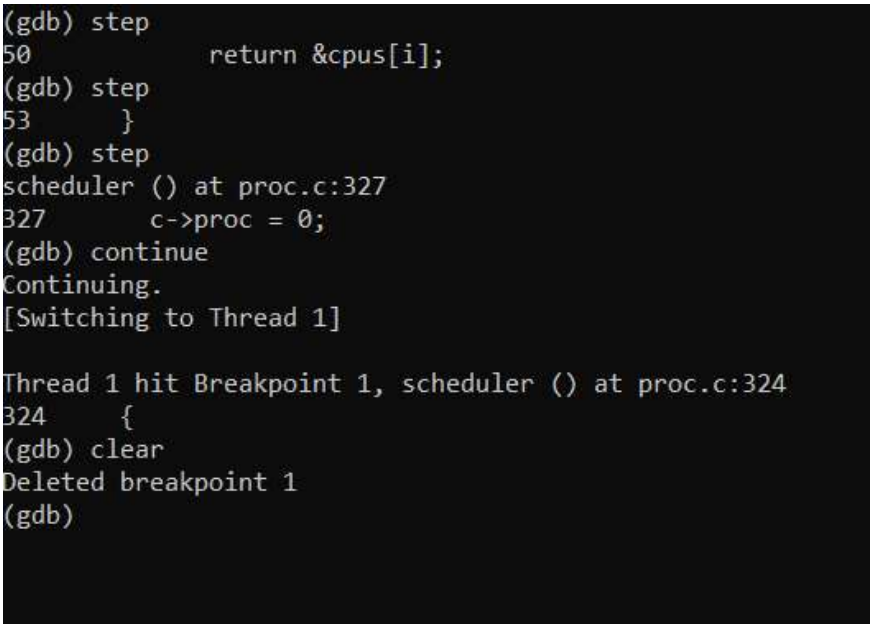
```
[Switching to Thread 2]

Thread 2 hit Breakpoint 2, exec (path=0x1c "/init", argv=0x8dfffed0) at exec.c:12
12      {
(gdb) continue
Continuing.

Thread 2 hit Breakpoint 2, exec (path=0x7ce "sh", argv=0x8dffeed0) at exec.c:12
12      {
(gdb) continue
Continuing.
[Switching to Thread 1]

Thread 1 hit Breakpoint 2, exec (path=0x1840 "ls", argv=0x8dfbeed0) at exec.c:12
12      {
(gdb) print argv[0]
$1 = 0x1840 "ls"
(gdb) print argv[1]
$2 = 0x1843 "-l"
(gdb) print argv[2]
$3 = 0x0
(gdb) backtrace
#0  exec (path=0x1840 "ls", argv=0x8dfbeed0) at exec.c:12
#1  0x80105380 in sys_exec () at sysfile.c:420
#2  0x80104837 in syscall () at syscall.c:139
#3  0x801058b9 in trap (tf=0x8dfbefb4) at trap.c:43
#4  0x8010561f in alltraps () at trapasm.S:20
#5  0x8dfbefb4 in ?? ()
Backtrace stopped: previous frame inner to this frame (corrupt stack?)
(gdb) up
#1  0x80105380 in sys_exec () at sysfile.c:420
420       return exec(path, argv);
(gdb) list
415           break;
416         }
417         if(fetchstr(uarg, &argv[i]) < 0)
418           return -1;
419       }
420       return exec(path, argv);
421     }
422
423     int
424     sys_pipe(void)
(gdb)
```

We made a breakpoint at swtch and continued to step through the code one line at a time. Then broke exec and went into a different thread. I was able to correctly recreate each of the steps the professor had instructed us to perform.

Scheduler Debug

```
mike@DESKTOP-SEEUKNP:~/cse460/temp1/xv6-public$ gdb
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
warning: File "/home/mike/cse460/temp1/xv6-public/.gdbinit" auto-loading has been declined by your `auto
-load safe-path' set to "$debugdir:$datadir/auto-load".
To enable execution of this file add
        add-auto-load-safe-path /home/mike/cse460/temp1/xv6-public/.gdbinit
line to your configuration file "/home/mike/.gdbinit".
To completely disable this security protection add
        set auto-load safe-path /
line to your configuration file "/home/mike/.gdbinit".
For more information about this security protection see the
"Auto-loading safe path" section in the GDB manual.  E.g., run from the shell:
        info "(gdb)Auto-loading safe path"
(gdb) target remote : 26000
Remote debugging using : 26000
0x0000fff0 in ?? ()
(gdb) file proc.c
A program is being debugged already.
Are you sure you want to change the file? (y or n) y
"/home/mike/cse460/temp1/xv6-public/proc.c": not in executable format: File format not recognized
(gdb) file proc
A program is being debugged already.
Are you sure you want to change the file? (y or n) y
proc: No such file or directory.
(gdb) file kernel
A program is being debugged already.
Are you sure you want to change the file? (y or n) y
Reading symbols from kernel...done.
(gdb) break scheduler
Breakpoint 1 at 0x80103a40: file proc.c, line 324.
(gdb) contine
Undefined command: "contine".  Try "help".
(gdb) continue
Continuing.
[Switching to Thread 2]
```

```
Thread 2 hit Breakpoint 1, scheduler () at proc.c:324
324     {
(gdb) step
326       struct cpu *c = mycpu();
(gdb) step
mycpu () at proc.c:42
42        if(readeflags()&FL_IF)
(gdb) step
readeflags () at x86.h:98
98        asm volatile("pushfl; popl %0" : "=r" (eflags));
(gdb) step
mycpu () at proc.c:42
42        if(readeflags()&FL_IF)
(gdb) step
45        apicid = lapicid();
(gdb) step
lapicid () at lapic.c:103
103       if (!lapic)
(gdb) step
102     {
(gdb) step
lapicid () at lapic.c:103
103       if (!lapic)
(gdb) step
105       return lapic[ID] >> 24;
(gdb) step
106     }
(gdb) step
lapicid () at lapic.c:105
105       return lapic[ID] >> 24;
(gdb) step
106     }
(gdb) step
mycpu () at proc.c:48
48        for (i = 0; i < ncpu; ++i) {
(gdb) step
49          if (cpus[i].apicid == apicid)
(gdb) step
48        for (i = 0; i < ncpu; ++i) {
(gdb) step
49          if (cpus[i].apicid == apicid)
(gdb) step
50            return &cpus[i];
(gdb) step
53      }
(gdb) step
50            return &cpus[i];
(gdb) step
```



Evaluation:

I was able to successfully complete each step in the lab. Modified msgs correctly and studied the IPCS commands. As well as used the debug in xv6 on both the swtch function and the scheduler function. I believe I have earned a 20/20 for this lab