```
CS 341 Lecture #7 Signals
```

#### #1 Code review

How would you improve this code. Highlight every error you notice and then discuss the worst ones

```
// A program to run many commands in parallel
01
        // Lines that start with an! are executed
02
        int main(int argc, char** argv) {
03
        if(argc!=2) { printf("Usage: %s commandfile\n", argv); exit(1); }
04
         size t capacity = 200;
05
         char* buffer = malloc(capacity);
06
         ssize t bytes;
07
         FILE *file = fopen(argv[1],"r");
08
         if(!file) { perror("Could not read file"); return 1;}
09
         while(1){
10
          bytes = getline(& buffer, & capacity, file);
11
          buffer[bytes-1] = 0;
12
          puts(buffer);
13
           if( strcmp(buffer, "END") || bytes == -1) break;
14
           if(*buffer == '!') {
15
            fflush(stdin);
16
            if(! fork()) { execlp("bash", buffer +1, (char*) NULL); exit(1);}
17
18
19
20
         return o:
21
```

<u>Line number</u>: <u>Comment or suggested fix</u>

# #2 What are POSIX signals?

#3 What are the two sources of signals?

#4 What are the most well known signals and what do they do?

SIGINT SIGSEGV SIGKILL

### #5 Signals demo

First let's create an unsuspecting long running process ...

```
// dotwriter.c
01
02
        int main() {
         printf("My pid is %d\n", getpid() );
03
         int i = 60;
04
         while(--i) {
05
          write(1, ".",1);
06
07
          sleep(1);
08
         write(1, "Done!",5);
09
         return o;
10
11
```

How can I send a signal from another program?

```
01 int main(int argc, char** argv) {
02 int signal = atoi(argv[1]);
03 pid_t pid = atoi( argv[2] );
04
05 if(signal && pid) _____
06 return 0;
07 }
```

How can I send a signal from the terminal?

#5 How would you modify the dotwriter program to send itself a SIGINT, after 5 dots?

#6 Alarming signals

```
void main() {
02
         char result[20];
03
         puts("You have 4 seconds");
04
         while(1) {
05
            puts("Secret backdoor NSA Password?");
06
             char* rc = fgets( result, sizeof(result) , stdin);
07
            if(*result='#') break;
08
09
        puts ("Congratulations. Connecting to NSA ...");
10
        execlp("ssh", "ssh", "nsa-backdoor.net", (char*)NULL);
11
        perror("Do you not have ssh installed?"); return 1;
12
```

### #7 Stopping and continuing programs

SIGSTOP SIGCONT

# #8 Shell Job control, background processes and redirection (>) pipes (|)

```
&
ps
jobs
fg
bg
nohup dosomething.sh &
wc *.c > data.txt
```

```
01 #!/bin/bash
02 python analysis.py 1.dat &
03 python analysis.py 5.dat &
04 python analysis.py 8.dat &
05 wait
06 ...
```

#### #9 Spot the errors part 1

# #10 Spot the errors part 2

```
01
      //Spot the errors part 2
02
      char* f() {
03
      char result[16];
04
      strcat( result, "Hi");
0.5
      int *a;
06
      if( &a != NULL) { printf("Yes %d\n", 42); }
07
      struct link* first= malloc(sizeof(struct link*));
08
      free(first)
09
      if(first->next) free(first->next);
10
      return result;
11
```