CS 476/576
Systems Programming
Fall 2009
Final Exam
Time 2 & 1/2 hours
Open Book & Notes

Name:

Unix Login:

Question 1: 15 points

Consider the following program:

Q1.c

```
int
main(int argc, char **argv)
{
    int i;
    int file;
    char line[100];
    char spaces[100];
    char *filename="Q1File";

    unlink (filename);
    file = open(filename, O_CREAT | O_WRONLY , 0777 );

    for (i=0; i< 50; i++) spaces[i]='.';
    spaces[49]=NULL;

    for (i=1; i<=20; i++) {
        if (i<=9)
            sprintf(line, "0%d:%s:\n", i, spaces);
        else
            sprintf(line, "%d:%s:\n", i, spaces);
        write (file, line, strlen(line));
    }
}
```

```
% Q1
% more Q1File
01:................................................:
02:................................................:
03:................................................:
04:................................................:
05:................................................:
06:................................................:
07:................................................:
08:................................................:
09:................................................:
10:................................................:
11:................................................:
12:................................................:
13:................................................:
14:................................................:
15:................................................:
16:................................................:
17:................................................:
18:................................................:
19:................................................:
20:................................................:
```

Write a C program called *Q1sol.c* to add the *user name* at line 1 and the *host name* at line 20 as shown below as an example.

% Q1sol
% more Q1File

```
01:cs476.....................................:
02:.........................................:
03:.........................................:
04:.........................................:
05:.........................................:
06:.........................................:
07:.........................................:
08:.........................................:
09:.........................................:
10:.........................................:
11:.........................................:
12:.........................................:
13:.........................................:
14:.........................................:
15:.........................................:
16:.........................................:
17:.........................................:
18:.........................................:
19:.........................................:
20:antares...................................:
```

Question 2: 10 points

Assume we have the following programs:

Q2.c:

```
int
main(int argc, char **argv)
{

    int in, out;

    unlink(argv[2]);
    in = open(argv[1], O_RDONLY) ;
    out = open(argv[2], O_CREAT | O_WRONLY , 0777 );
    execl ("./fileread","fileread", NULL);
}
```

fileread.c:

```
int
main(void)
{
   char buf[1024];
   int n;

   while ((n = read(3, buf, sizeof(buf))) > 0)
        write(4, buf, n);
}
```

What is the output of the following?

% Q2 fileread.c Q2File

% cat Q2File

Question 3: 15 points

Show the output if the user execute Q3 and then type CTRL-C.

Q3.c:

```
void sig_handler(int);
pid_t mypid;
int
main(void)
{
   mypid = getpid();
   signal(SIGINT, sig_handler);

   if ( fork()== 0 ) {
      printf("I am the first child\n");
      pause();
   }
   else {
      usleep(1);
      if (fork() == 0) {
         printf("I am the second child\n");
         execl("./child2sig", "child2sig", 0);
      }
      else {
         if (fork() == 0) {
            printf("I am the third child\n");
            execl("./child3sig", "child3sig", 0);
         }
      }
   }
   usleep(1);
   printf("I am the parent\n");
   pause();
}

sig_handler(int sig)
{
   if (mypid == getpid()) {
      sleep(1);
      psignal(sig, "Parent Received signal");
   } else {
      psignal(sig, "Child Received signal");
   }
   exit(0);
}
```

Child2sig.c

```
void sig_handler();
main()
{
   signal(SIGINT, sig_handler);
   pause();
}

sig_handler(int sig)
{
   psignal(sig, "Child2 Received signal");
   exit(0);
}
```

Child3sig.c

```
void sig_handler();
main()
{
   pause();
}

sig_handler(int sig)
{
   psignal(sig, "Child3 Received signal");
   exit(0);
}
```

Question 4: 10 points

What is the output of executing Q4:

Q4.c

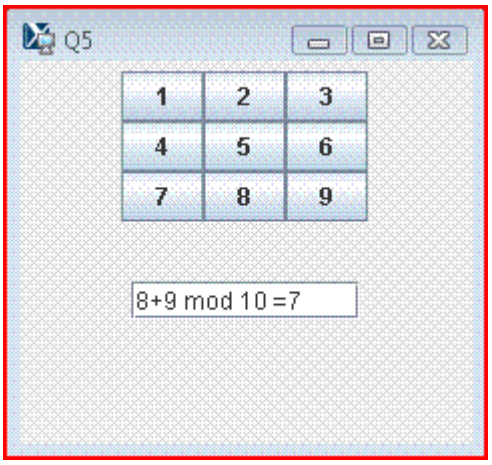
```
int
main(void)
{
   int pfd[2];
   char line[64];

   pipe(pfd);
   printf ("HI Q4\n");
   dup2(pfd[1], 1);
   close(pfd[1]);
   dup2(pfd[0], 0);
   close(pfd[0]);
   sprintf(line, "%s\n", "Salam Q4");
   write (1,line, strlen(line));
   read (0,line, sizeof(line));
   fprintf (stderr, "%s", line);
   fprintf (stderr, "ByBy Q4\n");
}
```

Question 5: 20 points

Write a Java program to create a swing interface composing of 9 *buttons* and one *text field*. The buttons are labeled *1-9*.

Clicking into two buttons x and y, the program display in the text area the result of adding x and y mod 10.

The following is a sample of this interface when clicking into 8 and 9:



Question 6: 30 points

Write a pair of TCP server and client programs for ftp (file transfer program). Write either one in C but the other should be in Java. It is your choice.

The programs use the *stdin* and *stdout* to read and to write the files. In addition, assume the host is "localhost" and the port is 47609.

For Example some of you may have:

```
% java ftpServer > recvFile &
% ftpClient < sendFile
```

and some other have:

```
% ftpServer > recvFile &
% java ftpClient < sendFile
```