PLEASE HAND IN

PLEASE HAND IN

UNIVERSITY OF TORONTO
Faculty of Arts and Science

DECEMBER EXAMINATIONS 2006

CSC 209H1 F
St. George Campus

Duration — 3 hours

Examination aids: One 8.5 x 11 sheet of paper

Student Number: |_|_|_|_|_|_|_|_|_|_|_|

Last Name: SOLUTIONS

First Name: _____

Instructor: _____

---

*Do **not** turn this page until you have received the signal to start.*
(In the meantime, please fill out the identification section above, and read the instructions below.)

This examination consists of 11 questions on 18 pages (including this one). *When you receive the signal to start, please make sure that your copy of the examination is complete and fill in your student number on every page.* If you need more space for one of your solutions, use the last pages of the exam or the back of this page and *indicate clearly the part of your work that should be marked.*

1: _____/ 8

2: _____/ 6

3: _____/ 8

4: _____/ 4

5: _____/ 9

6: _____/ 8

7: _____/ 8

8: _____/ 8

9: _____/10

10: _____/10

11: _____/10

TOTAL: _____/89

Good Luck!

Question 1. [8 MARKS]

Part (a) [5 MARKS]

Circle the correct answer below.

TRUE **[FALSE]** If we close the write end of a pipe, it can be reopened later as long as the read end of the pipe is still open.

TRUE **[FALSE]** A character array must always reserve space for the null termination character.

[TRUE] FALSE If a client's request for a socket connection arrives before the server has called `listen`, the connection will be denied.

[TRUE] FALSE If an array is passed in as an argument in a C function, any modifications made to the contents of the array will be visible when the function returns.

TRUE **[FALSE]** A static variable defined outside of a function is initialized once and then treated as a constant.

Part (b) [3 MARKS]

Given the following code, write the output of each of the labeled `printf` statements. If an error would occur, state what the error is.

```
char *str[4];
char *ptr;
char a[30] = "pointers are no fun";
str[0] = a;
str[1] = strchr(a, ' ') + 1;
str[2] = str[1] + 4;
ptr = strchr(str[1], ' ');
printf("%s %s %s\n", str[0], str[1], str[2]); /* A */
```

Output: pointers are no fun are no fun no fun

```
*(str[1] - 1) = '\0';
*ptr = '\0';
printf("%s %s %s\n", str[0], str[1], str[2]); /* B */
```

Output: pointers are no fun

```
int i;
for(i = 0; i < strlen(str[2]) + 1; i++) {
    str[2][i] = str[2][i + 3];
}
printf("%s %s %s\n", str[0], str[1], str[2]); /* C */
```

Output: pointers are fun

Question 2. [6 MARKS]

The current working directory contains 3 files: `list1`, `list2`, and `list3`. The contents of each file are shown below:

`list1`
```
santa naughty
elf nice
```

`list2`
```
rudolf nice
frosty nice
```

`list3`
```
dasher nice
elf naughty
```

I also have the following program called `status`

```
#!/bin/sh
person=$1
shift
for list in "$@"; do
   grep $person $list | cut -d " " -f 2
done
```

For each of the following lines, give the output that would appear when the command is run. Note that `set` does not produce output. Also note that `uniq -c` prints the number of times a line occurs with the contents of each unique line. (' is a single quote and ‘ is a back quote.)

```
cmd="status"
```

```
echo $cmd santa list1 *
```
status santa list1 list1 list2 list3

```
echo "$cmd  elf *"
```
status elf *

```
echo '$cmd rudolf *'
```
$cmd rudolf *

```
echo ‘$cmd rudolf *‘
```
nice

```
set ‘cut -d " " -f 1 list1
$cmd $1 list?
```
naughty

```
cut -d " " -f 2 list? | sort | uniq -c
```
2 naughty
4 nice

Question 3. [8 MARKS]

For each of the following calls to `strncpy`, indicate whether the third argument will produce the safe results regardless of the contents of either character array and regardless of the values of `SIZE1` and `SIZE2`. If the third argument might lead to a problem, describe the problem and provide values for `str1`, `str2`, `SIZE1`, and `SIZE2` to illustrate the problem.

```
char str1[SIZE1];
char str2[SIZE2];
```

A	`strncpy(str1, str2, strlen(str1));` We don't really care what the strlen of str1 is. It might be 0, but we can still copy str2 into str1.
B	`strncpy(str1, str2, strlen(str2));` If the strlen of str2 is bigger than the spacce allocated for str1, then it will overwrite the end of the array.
C	`strncpy(str1, str2, SIZE1);` Correct. If str2 is bigger than str1, then only the first part will be copied.
D	`strncpy(str1, str2, SIZE2));` If SIZE1 ¡ SIZE2 then we might end up overwriting the end of the array.

Question 4. [4 MARKS]

Part (a) [2 MARKS]

I frequently want to change to the course directory, but it is a long path to type each time. I wrote the following shell script to do this for me, but even though the script runs without errors, my current working directory is still the same. Explain why.

```
#!/bin/sh
cd /u/csc209h/fall/pub/
```

The cd command is run in a subshell. In other words a new process is forked and a copy of the shell is exec'd so the cd command is run in a different process. When that process terminates, all of the state associated with it is lost.

Part (b) [1 MARK]

I have a program that needs to be compiled differently on different machines. There is an environment variable called `OSTYPE` that describes the operating system on which you are working: `linux`, `darwin8.0`, or `solaris`. So I wrote three Makefiles called `Makefile.linux`, `Makefile.darwin8.0` and `Makefile.solaris`. Write a one line script that will run the appropriate Makefile. Recall that the `-f` option to `make` allows you to specify which file `make` will execute.

```
make -f Makefile.$OSTYPE
```

Part (c) [1 MARK]

A student came to my office and told me her "program crashed". What did she really mean?

She meant that it terminated abnormally. In other words it did not run to completion. Some error caused the program to terminate.

Question 5. [9 MARKS]

The following struct is used in all parts of this question:

```
struct node {
    int data;
    struct node *next;
};
```

Part (a) [3 MARKS]

Given the following statements, fill in the value for each of the expressions in the table below. If there is an error explain what the error is.

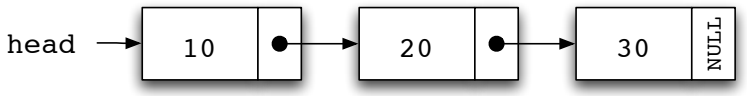
```
struct node a, b;
struct node *c;

a.data = 20; a.next = NULL;
```

`b = a;` `b.data = 10;`	`a.data ==` *20*
`c = a;` `c->data = 30`	`c->data ==` *Incompatible types in assignment*
`c = &a;` `c->data = 30`	`a.data ==` *30*

Part (b) [3 MARKS]

Write a C code fragment with the necessary declarations to produce a list that has the following structure.



Part (c) [3 MARKS]

Implement the linked list search method below.

```
/* Return a pointer to the first node containing value if value
 * is in the list pointed to by head.  Return NULL if value is not
 * in the list referred to by head.
 */
struct node *search(struct node *head, int value) {

    struct node *cur = head;

    while(cur != NULL) {
        if(cur->data == value) {
            return cur;
        }
    }
    return NULL;
}
```

Marking:

-

Question 6. [8 MARKS]

Part (a) [3 MARKS]

Write a C program fragment that includes the `printf` statement below to produce the following lines of output, although not necessarily in the order shown. No other output statement may be used. (The relationship between the pid numbers is important, not their actual values.)

```
printf("%d pid = %d ppid = %d\n", i, getpid(), getppid());
```

```
5 pid = 14319 ppid = 14318
4 pid = 14318 ppid = 14317
3 pid = 14317 ppid = 14316
2 pid = 14316 ppid = 14315
1 pid = 14315 ppid = 14310
0 pid = 14310 ppid = 1
```

```
int i;
int pid;
for(i = 0; i < 5; i++) {
    if((pid = fork()) > 0) {
         break;
    }
}
wait(NULL);
printf("%d pid = %d ppid = %d\n", i, getpid(), getppid());
```

Part (b) [1 MARK]

Is it possible to guarantee the order of the output in part a) using a single `wait` statement? If yes, put the `wait` statement in above. If no, explain why not.

Yes

Part (c) [3 MARKS]

Write a C program fragment that includes the `printf` statement below to produce the following lines of output, although not necessarily in the order shown. No other output statement may be used. (The relationship between the pid numbers is important, not their actual values.)

```
printf("%d pid = %d ppid = %d\n", i, getpid(), getppid());
```

```
0 pid = 14464 ppid = 14463
1 pid = 14465 ppid = 14463
2 pid = 14466 ppid = 14463
3 pid = 14467 ppid = 14463
4 pid = 14468 ppid = 14463
5 pid = 14463 ppid = 13984
```

```
int i;
int pid;
for(i = 0; i < 5; i++) {
    if((pid = fork()) == 0) {
         break;
    }
    wait(NULL);
}
printf("%d pid = %d ppid = %d\n", i, getpid(), getppid());
```

Part (d) [1 MARK]

Is it possible to guarantee the order ofs the output in part c) using a single `wait` statement? If yes, put the `wait` statement in above. If no, explain why not.

Yes

Question 7. [8 MARKS]

Part (a) [2 MARKS]

The program `elevator` returns an integer indicating which floor the elevator is on. Write a Bourne shell program that prints "Ground floor" if the program `elevator` returns 0.

```
elevator

if [ $? = 1 ]
then
    echo "Ground floor"
fi
```

Part (b) [6 MARKS]

I would like to give some of my files appropriate file extensions. Write a Bourne shell program that will rename files that are given as command line arguments. If a file is a Bourne shell program, it will be renamed to have the extension `.sh`. If a file is a plain text file, it will be renamed to have the extension `.txt`. If it is some other type, then it will not rename the file. Your program must not print any output if it is successful.

Recall that the `file` program prints the type of the file to standard output. In particular, if we run `file shprog` where `shprog` is a shell program then the output will be "shprog: Bourne shell script text executable". If we run `file textfile` the output will be "textfile: ASCII text".

```
#!/bin/sh

for file in "$@"
do
    type=‘file $file‘
    echo $type | grep Bourne > /dev/null 2>&1
    if [ $? -eq 0 ]
    then
        mv $file $file.sh
        continue
    fi

    type=‘file $file‘
    echo $type | grep ASCII> \dev\null 2>&1
    if [ $? -eq 0 ]
    then
        mv $file $file.txt
        continue
    fi
done
```

There is another obvious solution using cut.

Question 8. [8 MARKS]

You find yourself working on a system that lacks the `wait` and `waitpid` system calls, but you want your parent process to wait until the child has terminated. (You don't care about the return value from `wait` or removing zombies.) You are only worried about the process coordination aspects of `wait()`.

Write a C program that creates a child to run the program that is specified by the command line arguments to the main program. Without using the `wait` or `waitpid` system calls, ensure that the parent prints "All done" after the child terminates (regardless of how long the child runs). For example, the output of the program with the arguments `echo hello` would be

```
hello
All done
```

```
int childexited = 0;

void
handle_child(int code) {
  /* Do nothing.*/
  childexited = 1;
}


int main(int argc, char **argv)
{
  struct sigaction sa;
  sa.sa_handler = handle_child;
  sa.sa_flags = 0;
  sigemptyset(&sa.sa_mask);
  sigaction(SIGCHLD, &sa, NULL);

  int pid;
  if((pid = fork()) == 0) { /* child */
    execvp(argv[1],  &argv[1]);
  } else if(pid < 0) {
    perror("fork");
    exit(1);
  } else { /* parent */

    while(childexited == 0) {
      printf("going to sleep\n");
      sleep(1);
      printf("slept waiting for child\n");
    }
  }
  printf("All done\n");

  return 0;
}
```

Marking:

- 2 - signal handler installed
- 1 - signal handler installed before the fork
- 2 - not possible for parent to hang
- 2 - parent blocks
- 1 - parent prints all done

Question 9. [10 MARKS]

The program below creates a new process to run a text encryption function `simple_encrypt()`. The parent process reads data from standard input, writes it to the pipe and then reads the encrypted version from the child and writes it to standard output. There are two files: `pipe_encrypt.c` and `simple_encrypt.h`.

There are many errors in this program. Identify the errors and fix them, if possible, or explain clearly how to fix them. Please clearly label each error that you find with a number. Assume that all appropriate header files are included.

```
/* simple_encrypt.h */
int incr = 13; /* Default encryption value. */

/* Encrypt a string by replacing each lower case character with the
 * character incr in places forward in the alphabet.  If the result
 * goes past 'z', then it wraps around to the beginning of the alphabet
 */

char *simple_encrypt(char *buf, int size)
{
  int i;

  for(i = 0; i < size; i++) {

    if(islower(buf[i])) {
      buf[i] = buf[i] + incr;
      if(buf[i] > 'z') {
        buf[i] -= 26;
      }
    }
  }
  return buf;
}
```

```
/* pipe_encrypt.c */
#include "simple_encrypt.h"
#define SIZE 128

extern int incr;

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

  int fd[2];
  char *line;
  int nr, num;

  /* If a command line argument was provided, use it as the
   * increment, otherwise use the default.
   */
  if(argc == 2) {
    incr = argv[1];
  }

  if(pipe(fd) == -1) {
    perror("pipe");
    exit(1);
  }

  if((num = fork()) == 0) {

    while((nr = read(fd[0], line, SIZE)) != 0) {

      simple_encrypt(line, SIZE);

      write(fd[1], line, SIZE);

    }
  } else {

    while(fgets(line, SIZE, stdin)) {
      write(fd[1], line, strlen(line));
    }


    while((read(fd[0], line, SIZE)) != 0) {
      printf("%s\n", line);
    }

  }
  return 0;
}
```

Question 10. [10 MARKS]

Write a C function that takes a directory name as an argument and prints both the number of files and the number of directories found in the directory. The function should return `-1` if the argument is not a directory, and `0` otherwise.

Recall that the `struct stat` contains a field `st_mode`. There are several bitmasks available to extract the type including `S_IFDIR` (directory) and `S_IFREG` (regular file).

```
int
numdirs(char *dirname) {

  DIR *dir;
  struct dirent *dp;
  struct stat sbuf;
  int dircount = 0;
  int filecount = 0;
  char path[128];

  if((dir = opendir(dirname)) == NULL) {
      return -1;
  }

  while((dp = readdir(dir)) != NULL) {
    strncpy(path, dirname, 128);
    strncat(path, "/", 128);
    strncat(path, dp->d_name, 128);
    if(stat(path, &sbuf) == -1) {
      perror("stat");
    } else {
      if(sbuf.st_mode & S_IFDIR) {
        dircount++;
      } else {
        filecount++;
      }
    }
  }
  printf("Number of directories = %d\n", dircount);
  printf("Number of files = %d\n", filecount);
  return 0;
}
```

Marking:

- 1 for function heading with correct return value and correct argument type
- 1 for opendir
- 1 for loop with readdir

- 1 for constructing the path for stat
- 2 for correctly calling stat with valid sbuf pointer
- 1 for getting the mode
- 1 for counting correctly
- 1 for printing
- 1 for error return value.

Question 11. [10 MARKS]

Complete the function `copy2files` below that concurrently copies data from two file descriptors (`fromfd1` and `fromdf2` to two different file descriptors (`tofd1` and `tofd2`). The function returns the total number of bytes read from the two file descriptors.

You solution must ensure that the program does not block waiting for data from one file descriptor when there is data available to read on the other file descriptor.

You may assume that the following function is available to you. Do not write the body of the function.

Note that there is more space on the next page.

```
/* Read a chunk of data from fromfd and write it to tofd.  Return the
 * number of bytes read. */
int readwrite(int fromfd, int tofd)

int copy2files(int fromfd1, tofd1, int fromfd2, int tofd2) {
```

Robbins and Robbings page 111

```
int copy2files(int fromfd1, int tofd1, int fromfd2, int tofd2) {
   int bytesread;
   int maxfd;
   int num;
   fd_set readset;
   int totalbytes = 0;

   if ((fromfd1 < 0) || (fromfd1 >= FD_SETSIZE) ||
       (tofd1 < 0) || (tofd1 >= FD_SETSIZE) ||
       (fromfd2 < 0) || (fromfd2 >= FD_SETSIZE) ||
       (tofd2 < 0) || (tofd2 >= FD_SETSIZE))
      return 0;
   maxfd = fromfd1;                    /* find the biggest fd for select */
   if (fromfd2 > maxfd)
      maxfd = fromfd2;

   for ( ; ; ) {
      FD_ZERO(&readset);
      FD_SET(fromfd1, &readset);
      FD_SET(fromfd2, &readset);
      if (((num = select(maxfd+1, &readset, NULL, NULL, NULL)) == -1) &&
         (errno == EINTR))
         continue;
      if (num == -1)
         return totalbytes;
      if (FD_ISSET(fromfd1, &readset)) {
         bytesread = readwrite(fromfd1, tofd1);
         if (bytesread <= 0)
            break;
```

```
            totalbytes += bytesread;
         }
         if (FD_ISSET(fromfd2, &readset)) {
            bytesread = readwrite(fromfd2, tofd2);
            if (bytesread <= 0)
               break;
            totalbytes += bytesread;
         }
      }
      return totalbytes;
}
```

Total Marks = 89