CSCE 510 - Systems Programming Fall 2005
Test 1
September 22, 2005

M Matthews

September 22, 2005

Name:____ Email:____

Instructions

- Make sure your exam is complete. There should be 12 pages including this cover sheet.
- No Calculators, cell phones, or other electronic devices.
- Answer in the space provided if at all possible.
- If a question is unclear please ask early in the test.
- There is a Take Home question. It will be emailed today.
- If you need to know any IA32 instruction just ask.
- Good Luck!

CSCE 510 - Systems Programming **Test-1 September 22, 2005**

1. CPP

   (a) Write a macro median(a,b,c) that will return the middle number of the three.

   (b) Show the steps in the transformation of putchar(x) to its final form. Show all intermediate steps. ("/usr/include/stdio.h" on last page of this test.)

2. Stdio - Throughout this problem assume the buffer size is 1024=1KB.

   (a) Describe each of the components of a “struct _iobuf”

   (b) What is _iob[2]?

   (c) What is the difference between a FILE and a file descriptor?

   (d) How many reads would be executed in reading a file of 1MB using the code below:

```
FILE *fp;
  ...
while((c = getc(fp)) != EOF)
   if (c == '\n') ++count;
```

3. File Systems

   (a) What are the four major sections of a file system and briefly describe what is in each section?

   (b) How would you find out the inode of “/bin”?

   (c) What is in an inode that is not returned by a stat system call?

   (d) What really happens when the command `ln ../a/b d` is executed?

   (e) What really happens when the command `rm f1` is executed?

4. Permissions

   (a) In addition to permissions what is in a mode_t?

   (b) Write a section of code that will test whether “file” has write permissions for others?

   (c) Explain is great detail how permissions are checked in opening “/a/b/c/d?”

5. Unix File Systems II - Suppose an inode is 128 bytes long, pointers are 4 butes long, and the status information takes up 68 bytes. Assume there are single, double and triple indirect pointers. Assume also that the block size is 8K?

   (a) How many direct block pointers will fit in such an inode?

   (b) How many pointers can be put in a block?

   (c) What is the size of the largest file that does not utilyze the single indirect pointer?

   (d) What is the size of the largest file that does not utilyze the triple indirect pointer?

   (e) Explain how the addition of a single byte to a file can cause Unix to allocate two additional data blocks

6. AR

   (a) What marks the end of an ar header?

   (b) Explain in detail how the compiler finds the library specified in `gcc  prog.c -o prog -lque`

   (c) Explain in detail how the compiler finds the library specified in `gcc  prog.c -o prog -L/my -lque`

   (d) On average how much space is wasted in "fragmentation" in storing a file?

   (e) So if an archive has 1000 files in it how much space is saved over using individual files.

   (f) Assuming you have `char *offsetString` how do you move to the indicated position in file, offsetting from the start of the file.

7. Time

   (a) Write a section of code that will change the modification time of a file to two days earlier than now.

   (b) Write a function `time_t newest(char *path)` that will return the newest file in the directory "path."

8. Fork

   (a) Show the standard fork skeleton.

   (b) What things are different in the Parent and child?

   (c) What happens to an open file when a process forks?

   (d) When a parent dies, the child becomes an orphan and is inherited by which process?

   (e) Write a small program which will fork and have the child print the group id of its parent while the parent prints the pid of the child?

   (f) Write a small program cause fork to fail.

9. Exec

   (a) Give an exec family call to execute "/bin/ls" passing arguments "-l" and "-t".

   (b) What actually happens when the system call execve is executed?

   (c) Give two reasons an execve system call could fail.

   (d) What happens when an exec call fails?

   (e) What is the major difference between execvp and execv?

   (f) What is the major difference between execl and execv?

10. Draw a diagram that details the Memory Layout of a Unix process. Then give an example (a C program) that has an element in each section.

```
#define stdin (&_iob[0])
#define stdout (&_iob[1])
#define stderr (&_iob[2])

#define NULL 0
#define EOF (-1)

#define getc(p) (--(p)->_cnt>=0? *(p)->_ptr++&0377:_filbuf(p))
#define getchar() getc(stdin)
#define putc(x,p) \
(--(p)->_cnt>=0? \
((int)(*(unsigned char *)(p)->_ptr++=(x))):\
_flsbuf((unsigned char)(x),p))
#define putchar(x) putc(x,stdout)
#define feof(p) (((p)->_flag&_IOEOF)!=0)
#define ferror(p) (((p)->_flag&_IOERR)!=0)
#define fileno(p) ((int)((p)->_file))
```