

All questions are equally weighted.

1. (a) Where is the free space-list (free data blocks) saved?

**ANSWER:** in the super-block

- (b) What function is used to make sure this data structure is correctly updated before events such as the system going down?

**ANSWER:** sync is the function, this was however not covered this semester so far.

- (c) What place(s) is(are) checked for permissions data when “/a/b/c” is opened for reading?

**ANSWER:**

- i. first in the inode of / the system check for the owner, group, others whichever is the first of these to apply
- ii. then in the inode of /a
- iii. then in the inode of /a/b
- iv. then in the inode of /a/b/c

2. AR implementation

- (a) What is the function of the magic string for ar archives?

**ANSWER:** the string “!  
n” identifies this as an archive file; in that it is unlikely that another file would start with this string.

- (b) What is the major difference in the structure of archives that contain binary files from those that don't?

**ANSWER:** You can't use more on the ones with binary files.

- (c) Assuming f1 is a new file, in performing “ar -r arch f1”, how does ar get the owner of the file “f1” and what steps does it have to do before writing the header?

**ANSWER:** use a “stat(“f1”, &statbuf);” system call

- (d) Supposing that one header had just been read, give code showing how move to the next “file header” in the archive file.

**ANSWER:** copy the string that represents the length with something like

```
strncpy(newbuf, buf[OffsetToLengthString], 6);           // copy 6 bytes
lsize = strtol (newbuf, NULL, 10);                       // convert to a long
// check to see if there is padding (i.e., is the length odd)
if (lsize % 2 == 1) ++lsize;                             // if so increment
if (lseek(fileDescriptor, lsize, SEEK_CUR) < 0)         // move to next header
```

### 3. CPP

- (a) define a macro MAX3(a,b,c) that returns the maximum of three numbers. If you like you can use MAX(a,b) that returns the maximum of 2 numbers.

**ANSWER:**

```
#define MAX(a,b) (((a) > (b)) ? (a) : (b)) //Why all the extra parentheses?  
#define MAX3(a,b,c) ((Max(a,b) > Max(b,c)) ? (Max(a,b)) : (Max(b,c)))
```

- (b) How many cpp transformations (macro expansions) are performed on the line  
while ((c = getchar()) != EOF)  
?

**ANSWER:** The pertinent macros are:

```
#define getchar() getc(stdin)  
#define getc(p) (--(p)->_cnt < 0 ? __filbuf(p) : (int)*(p)->_ptr++)  
#define stdin (&_iob[0])
```

So then the transformation sequence is:

```
getchar()  
⇒ getc(stdin)  
⇒ (--(stdin)->_cnt < 0 ? __filbuf(stdin) : (int)*(stdin)->_ptr++)  
⇒ (--(&_iob[0])->_cnt < 0 ? __filbuf(stdin) : (int)*(stdin)->_ptr++)  
⇒ (--(&_iob[0])->_cnt < 0 ? __filbuf(&_iob[0]) : (int)*(stdin)->_ptr++)  
⇒ (--(&_iob[0])->_cnt < 0 ? __filbuf(&_iob[0]) : (int)*(&_iob[0])->_ptr++)
```

- (c) What is the result of the first transformation?

**ANSWER:**

- (d) What is the result of the next transformation (if there is a next one)?

**ANSWER:**

### 4. Memory Layout

- (a) Show the memory layout of a UNIX process.

**ANSWER:** See notes.

**ANSWER:**

- (b) Give a section of code that will get the SHELL variable from the environment.

**ANSWER:**

```
char *envString;  
:  
envString = getenv('SHELL');
```

- (c) What happens to the memory layout when you add a couple of items to the environment? (i.e., what moves where?)

**ANSWER:** The table of pointers needs to be moved to the heap.

The pointers to the original strings still point to the original locations but the table is now on the heap and hence the environ pointer points to the heap.

5. Given that the program shown below

```
#include <stdio.h>
int *p = NULL;
void a (int );

main(){
    int n;
    n = 4;
    a(n);
}

void a (int i)
{
    int j = i;
    if(j > 0){
        if (p == NULL) p = &j;
        else{
            printf("the size is %d\n", p-&j);
            p = &j;
        }
        a(j-1);
    }
}
```

produces the output  
the size is 30  
the size is 30  
the size is 30

What is 30 the size of?  
\vfill  
{\tt {\bf ANSWER}: }  
\vfill

What elements (data items) are in that area?  
\vfill  
{\tt {\bf ANSWER}: }  
\vfill

6. Assume the UFS format for this question.

- (a) Show a diagram and explain how adding one byte to a file might add three data blocks to the file's space requirements.

**ANSWER:** Diagram no way! But this occurs when you

- filled all the direct blocks (10)
- filled all the data blocks pointed to by the single indirect pointers

Then adding one more character will caused the allocation of

- i. the double indirect data block of pointers
- ii. the first data block of indirect pointers that is pointed to by this block
- iii. and the data block for the new character.

- (b) Assuming that blocks are 8K and that each block can hold 2048 pointers. At which size would you expect for adding one character to a file add three blocks. Leave you answer as an expression, such as  $45 * 8K + 8K * 8K + 47$ .

**ANSWER:**

10 direct blocks of 8K bytes each + 2048 single indirect DBs of 8K bytes each + 1  
=  $10 * 8K + 2048 * 8K + 1$

7. Write a complete program that when run will create another process and the child process print out a message containing its parent's process ID and the parent process should print out the child's pid.

8. GDB

- (a) How do you compile so that you can use gdb to debug your programs?

**ANSWER:** Use the `-g` option

- (b) How do you set a breakpoint on line 35?

**ANSWER:** `b 35`

- (c) How do you single step (stepping into functions)?

**ANSWER:** use the command `'s'`

- (d) How do you show the stack of activation records that is active?

**ANSWER:** `'backtrace'` or `'bt'`

**ANSWER:**

9. Write a C program "newerthan t f1 f2 ... fn", that takes an integer t(=time) in days and a list of files f1 f2 ... fn and prints the name of those files that were created since t days ago.

**ANSWER:**

10. TAKE-HOME due Tuesday Oct 2 Midnight

Make a directory named Test1 and include in it a Makefile, name the executable "test1" and have clean and test targets, also.

- (a) **Undergraduate** Write a program that will take either one or no arguments. If there is an argument it should be a path to a directory otherwise the current directory (“.”) is used. This program should find the total number of blocks used by all of the files in this directory. You may implement this so that it does not recursively enter subdirectories. *For extra credit you can recursively count subdirectories.*
- (b) **Graduate** Write a function `getelement()` that allocates space for an “element”, reads and initializes the element from the stdin and returns a pointer to the element. An element is a structure containing an integer “val” and “next,” a pointer to the next element in the list. Also assume that a global pointer “head” points to a list of elements that are to be re-processed much as in the manner that `ungetch-getch` (or `ungetc-fgetc`) reprocess characters. Write a function `nextelem` that checks head and if it is not null returns a pointer to the first element of the list. If head is null then `getelement` is called and the pointer returned from the call is the value returned as the value of `nextelem`.
- Write a test program that tests this function.

Send questions on the Take-Home via email. I will be in some on Saturday and/or Sunday. **Read email! because I respond to everyone.**