

Week 09 Tutorial Questions

1. The assignment specification doesn't fully explain the assignment. What can I do?
2. What is `gitlab.cse.unsw.edu.au`? How do I want use it for assignment 2?
3. Discuss strategies for decoding instructions for the assignment.

Assembler	C	Bit Pattern
<code>add \$d, \$s, \$t</code>	<code>d = s + t</code>	000000ssssstttttddddd00000100000
<code>sub \$d, \$s, \$t</code>	<code>d = s - t</code>	000000ssssstttttddddd00000100010
<code>and \$d, \$s, \$t</code>	<code>d = s & t</code>	000000ssssstttttddddd00000100100
<code>or \$d, \$s, \$t</code>	<code>d = s t</code>	000000ssssstttttddddd00000100101
<code>slt \$d, \$s, \$t</code>	<code>d = 1 if s < t else 0</code>	000000ssssstttttddddd00000101010
<code>mul \$d, \$s, \$t</code>	<code>d = s * t</code>	011100ssssstttttddddd00000000010
<code>beq \$s, \$t, I</code>	<code>if (s == t) PC += I</code>	000100ssssstttttIIIIIIIIIIIIIIIIII
<code>bne \$s, \$t, I</code>	<code>if (s != t) PC += I</code>	000101ssssstttttIIIIIIIIIIIIIIIIII
<code>addi \$t, \$s, I</code>	<code>t = s + I</code>	001000ssssstttttIIIIIIIIIIIIIIIIII
<code>slti \$t, \$s, I</code>	<code>t = (s < I)</code>	001010ssssstttttIIIIIIIIIIIIIIIIII
<code>andi \$t, \$s, I</code>	<code>t = s & I</code>	001100ssssstttttIIIIIIIIIIIIIIIIII
<code>ori \$t, \$s, I</code>	<code>t = s I</code>	001101ssssstttttIIIIIIIIIIIIIIIIII
<code>lui \$t, I</code>	<code>t = I << 16</code>	00111100000tttttIIIIIIIIIIIIIIIIII
<code>syscall</code>	<code>syscall</code>	000000000000000000000000000000001100

4. What should my `smips.c` do for this example:

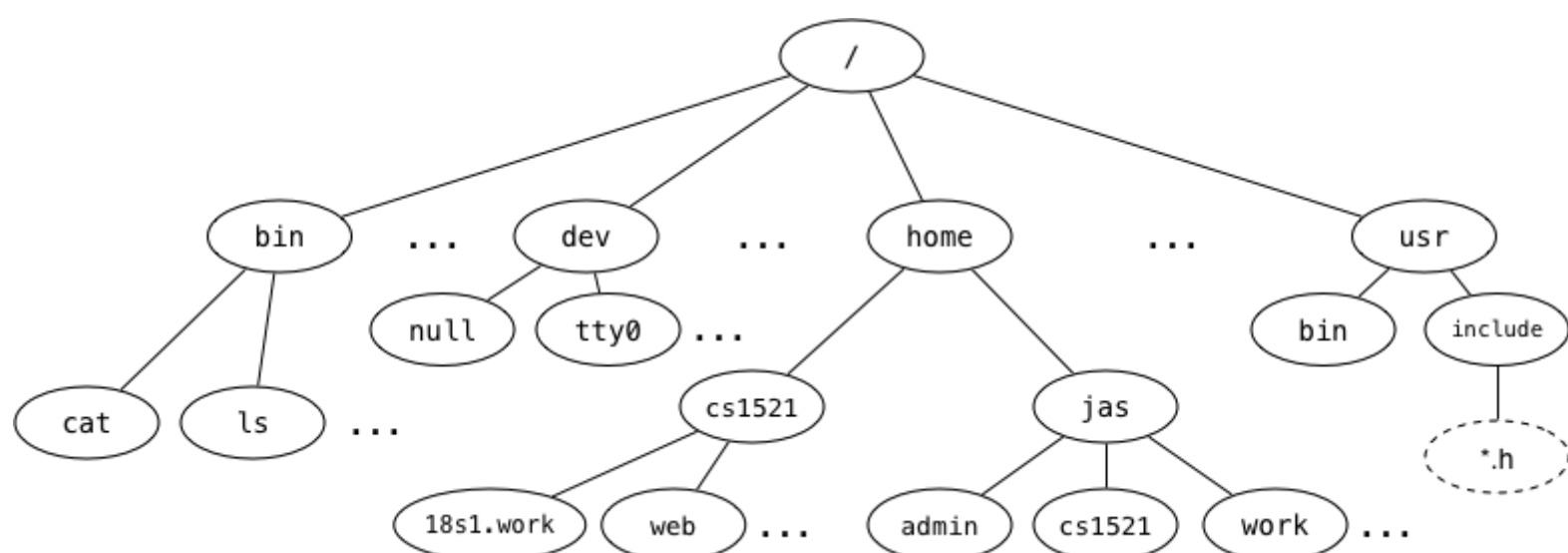
```
$ cat test.s
136bffa
$ ./smips test.s
```

Decode it this instruction mean?

What should happen when t is executed by `smips.c`

How can check my answer?

5. We say that the Unix filesystem is *tree-structured*, with the directory called `/` as the root of the tree, e.g.,



Answer the following based on the above diagram:

- a. What is the full pathname of COMP1521's web directory?
- b. Which directory is `~jas/../../..`?
- c. Links to the children of a given directory are stored as entries in the directory structure. Where is the link to the parent directory stored?
- d. What kind of filesystem object is `cat`?
- e. What kind of filesystem object is `home`?

- f. What kind of filesystem object is `tty0`?
 - g. What kind of filesystem object is a symbolic link? What value does it contain?
 - h. Symbolic links change the filesystem from a tree structure to a graph structure. How do they do this?
6. The `stat()` and `lstat()` functions both take an argument which is a pointer to a `struct stat` object, and fill it with the meta-data for a named file.

On Linux, a `struct stat` contains the following fields (among others, which have omitted for simplicity):

```
struct stat {
    ino_t st_ino;          /* inode number */
    mode_t st_mode;        /* protection */
    uid_t st_uid;          /* user ID of owner */
    gid_t st_gid;          /* group ID of owner */
    off_t st_size;         /* total size, in bytes */
    blksize_t st_blksize; /* blocksize for filesystem I/O */
    blkcnt_t st_blocks;    /* number of 512B blocks allocated */
    time_t st_atime;       /* time of last access */
    time_t st_mtime;       /* time of last modification */
    time_t st_ctime;       /* time of last status change */
};
```

Explain what each of the fields represents (in more detail than given in the comment!) and give a typical value for a regular file which appears as follows:

```
$ ls -ls stat.c
8 -rw-r--r-- 1 jas cs1521 1855 Sep 9 14:24 stat.c
```

Assume that `jas` has user id 516, and the `cs1521` group has group id 36820.

7. Consider the following (edited) output from the command `ls -l ~cs1521`:

```
drwxr-x--- 11 cs1521 cs1521 4096 Aug 27 11:59 17s2.work
drwxr-xr-x 2 cs1521 cs1521 4096 Aug 20 13:20 bin
-rw-r----- 1 cs1521 cs1521 38 Jul 20 14:28 give.spec
drwxr-xr-x 3 cs1521 cs1521 4096 Aug 20 13:20 lib
drwxr-x--x 3 cs1521 cs1521 4096 Jul 20 10:58 public_html
drwxr-xr-x 12 cs1521 cs1521 4096 Aug 13 17:31 spim
drwxr-x--- 2 cs1521 cs1521 4096 Sep 4 15:18 tmp
lrwxrwxrwx 1 cs1521 cs1521 11 Jul 16 18:33 web -> public_html
```

- a. Who can access the `17s2.work` directory?
 - b. What operations can a typical user perform on the `public_html` directory?
 - c. What is the file `web`?
 - d. What is the difference between `stat("web", &info)` and `lstat("web", &info)`? (where `info` is an object of type `(struct stat)`)
8. Write a C program, `chmod_if_public_write.c`, which is given 1+ command-line arguments which are the pathnames of files or directories
- If the file or directory is publically-writeable, it should change it to be not publically-writeable, leaving other permissions unchanged.

It also should print a line to `stdout` as in the example below

```
$ gcc chmod_if_public_write.c -o chmod_if_public_write
$ ls -ld file_modes.c file_modes file_sizes.c file_sizes
-rwxr-xrwx 1 z5555555 z5555555 116744 Nov 2 13:00 file_sizes
-rw-r--r-- 1 z5555555 z5555555 604 Nov 2 12:58 file_sizes.c
-rwxr-xr-x 1 z5555555 z5555555 222672 Nov 2 13:00 file_modes
-rw-r--rw- 1 z5555555 z5555555 2934 Nov 2 12:59 file_modes.c
$ ./file_modes file_modes file_modes.c file_sizes file_sizes.c
removing public write from file_sizes
file_sizes.c is not publically writable
file_modes is not publically writable
removing public write from file_modes.c
$ ls -ld file_modes.c file_modes file_sizes.c file_sizes
-rwxr-xr-x 1 z5555555 z5555555 116744 Nov 2 13:00 file_sizes
-rw-r--r-- 1 z5555555 z5555555 604 Nov 2 12:58 file_sizes.c
-rwxr-xr-x 1 z5555555 z5555555 222672 Nov 2 13:00 file_modes
-rw-r--r-- 1 z5555555 z5555555 2934 Nov 2 12:59 file_modes.c
```

Make sure you handle errors.

9. Write a C program, `farep.c`, which is given 1+ command-line arguments which is a string to search for.

If there is only 1 command-line argument it should read lines from stdin and print them to stdout iff they contain the string specified as the first command line argument.

If there are 2 or more command line arguments, it should treat arguments after the first as filenames and print any lines they contain which contain the string specified as the first command line arguments.

When printing lines your program should prefix them with a line number.

It should print suitable error messages if given an incorrect number of arguments or if there is an error opening a file.

COMP1521 20T2: Computer Systems Fundamentals is brought to you by
the [School of Computer Science and Engineering](#)
at the [University of New South Wales](#), Sydney.
For all enquiries, please email the class account at cs1521@cse.unsw.edu.au

CRICOS Provider 00098G