**Threading**

**Pthread\_t** = thread ‘type’

**Pthread\_create**(&pthread\_t, NULL, functionPointer, argStruct|NULL); (Returns non-zero on fail)

**Pthread\_join**(pthread\_t, void\*\* returnVal) = {like wait\_pid()}

**Pthread\_self()** {returns thread id of the current thread}

**Pthread\_exit**() {exits the current thread}

**Pthread\_detach**(pthread\_t) {detaches from parent for init to reap immediately on exit}

Mutex = binary semaphore used for mutual exclusion

**Sem\_t** {type used for semaphore functions}

**Sem\_init**(&sem\_t, 0, initval);

**Sem\_wait**(&sem\_t); ---- alias: wait()

**Sem\_post**(&set\_t); ---- alias: signal()

**Sem\_destroy**(&sem\_t);

**Sem\_open**(“location”, O\_CREAT, initval); {named semaphore, no need for shared memory}

**Sem\_unlink**(“location”); {destroy named semaphore}

**Sem\_getvalue**() – return value of semaphore

**Sem\_trywait**() – don’t block if semaphore value is zero

**Sem\_timewait**() – wait until timeout

**Pthread\_mutex\_t**()

**Pthread\_mutex\_init**()

**Pthread\_mutex\_lock**()

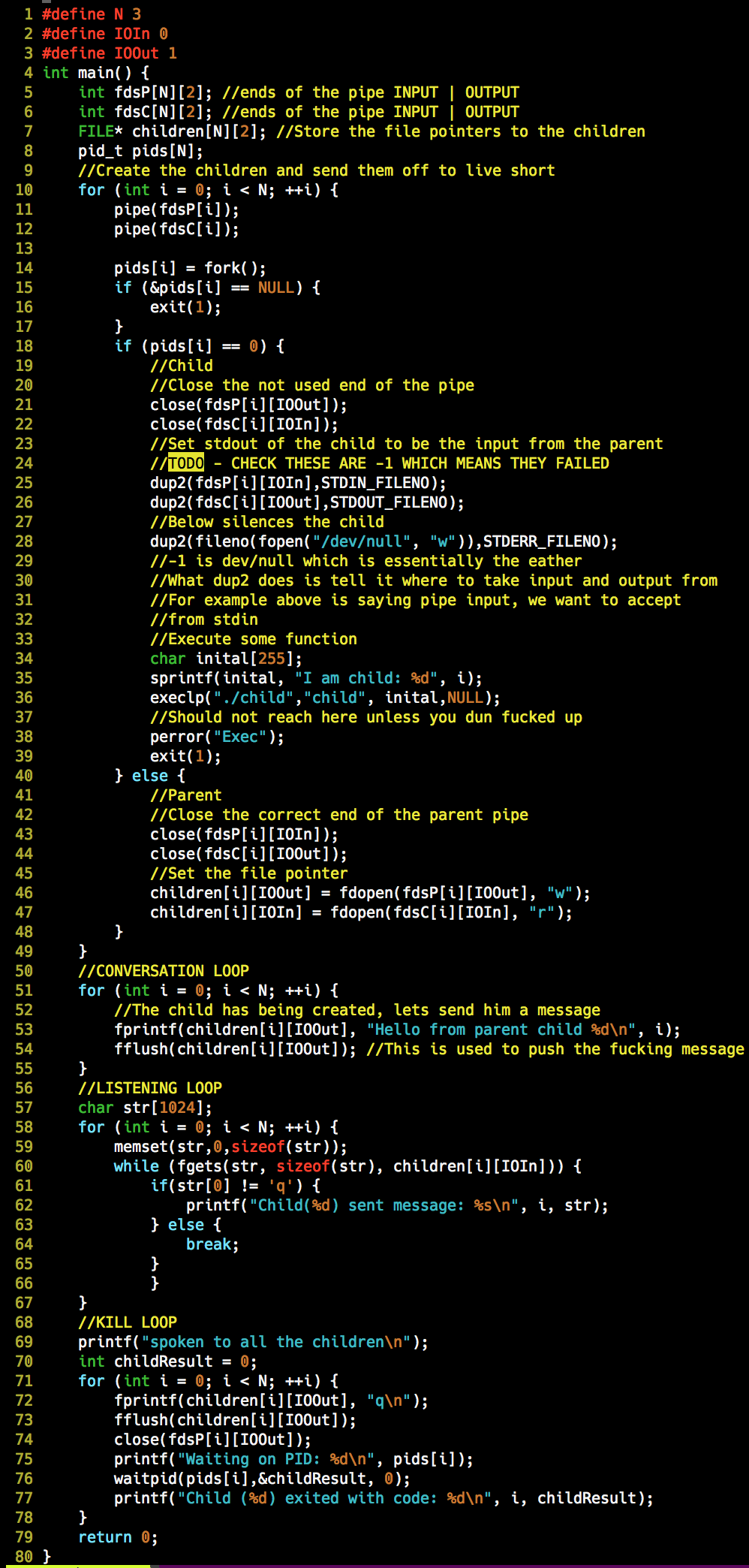
**Pthread\_mutex\_unlock**()

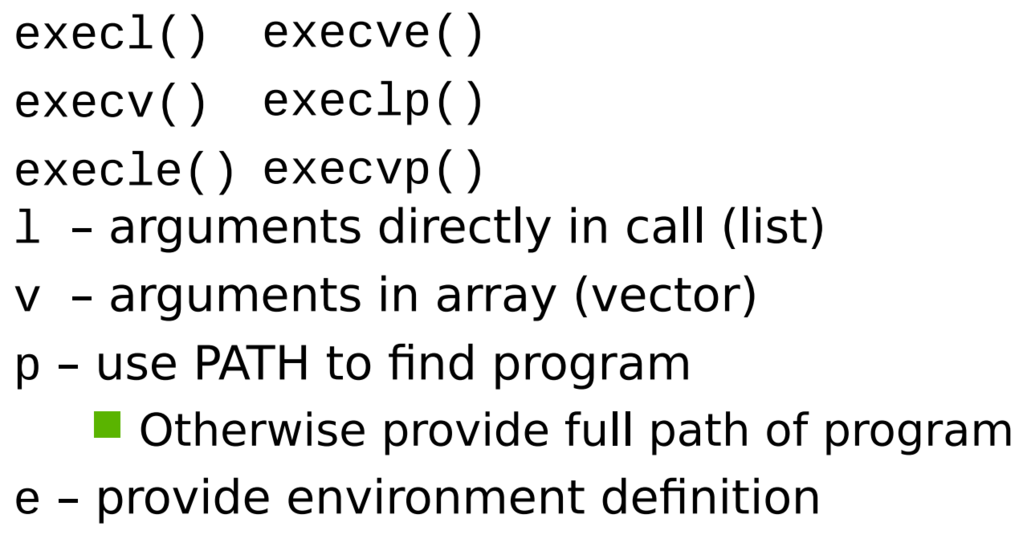
Threadsafe functions:

Pread()

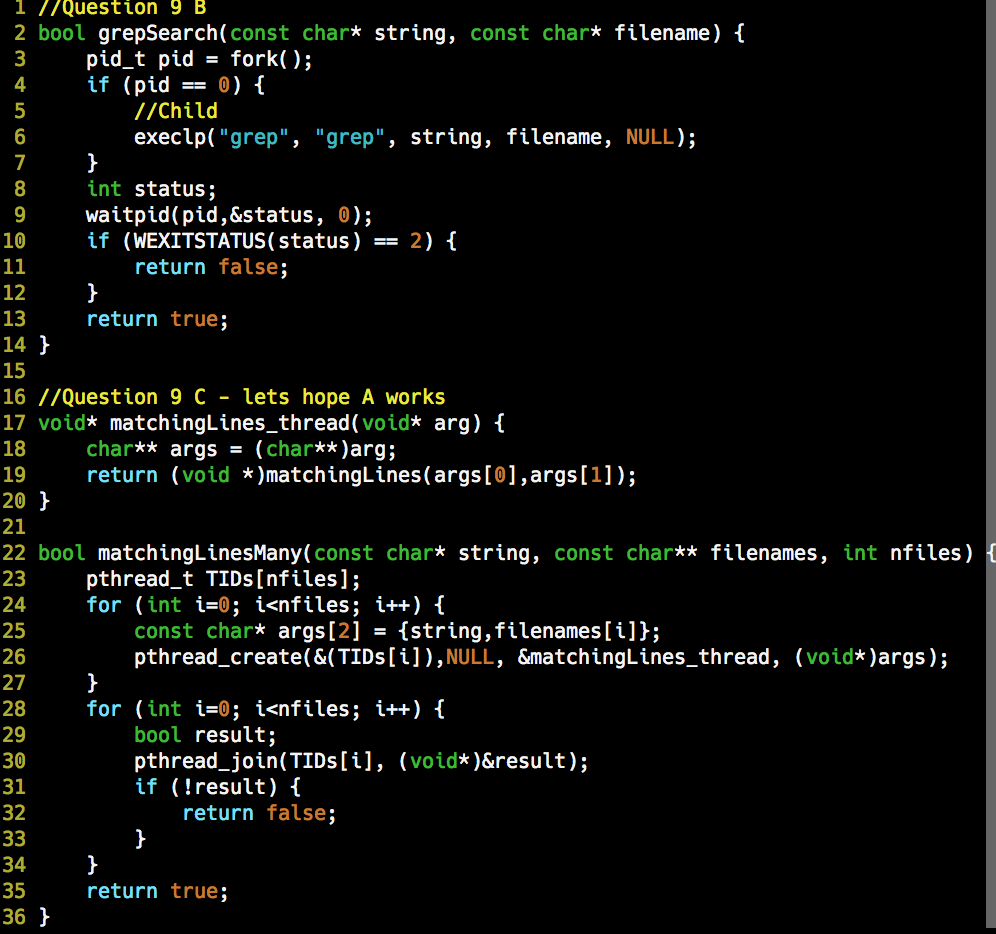
Localtime\_r()

**Process/Pipe/Dup2/File Printing/Reaping**

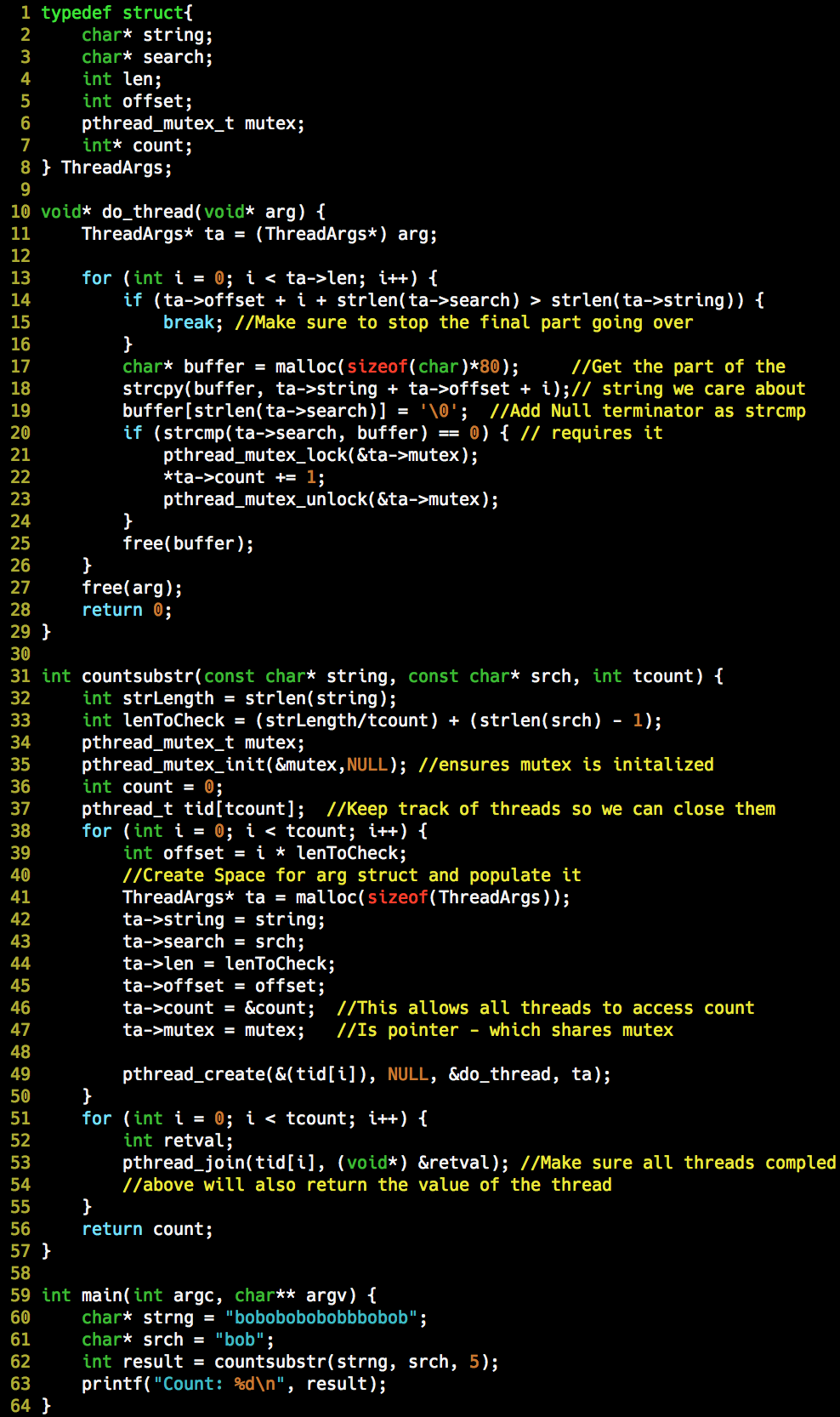




**Basic Process & Threading**



**Threading**



**File Access**

FILE\* fopen(‘path’, ‘r|w|a’); {if return value is NULL, error opening}

char\* fgets(buffer, maxcharcount, FILE\*) {returns NULL if there’s an error, otherwise pointer to char\* buffer}

int fclose(FILE\*) {closes the file stream, returns EOF on error, or zero if successful}

**string shit**

char **strstr**(char\* haystack, char\* needle) {returns pointer to the first occurance or NULL}

**strcmp**(str1, str2) returns <0 if str1 < str2; 0 if identical.

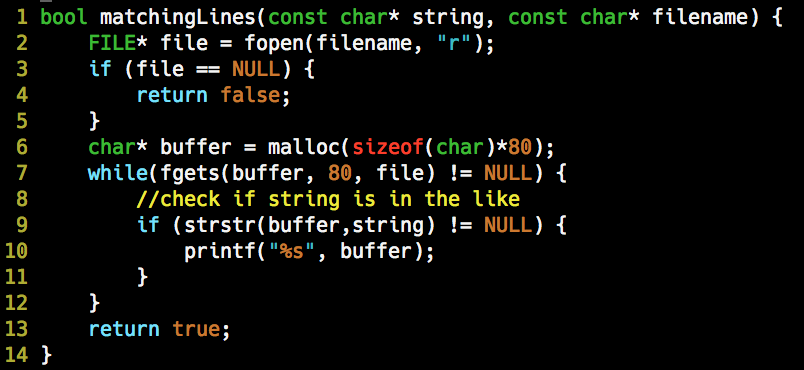
**strcpy**(destination, “string”, sizeof(destination))

**strcat**(char\* dest, const char\* src);

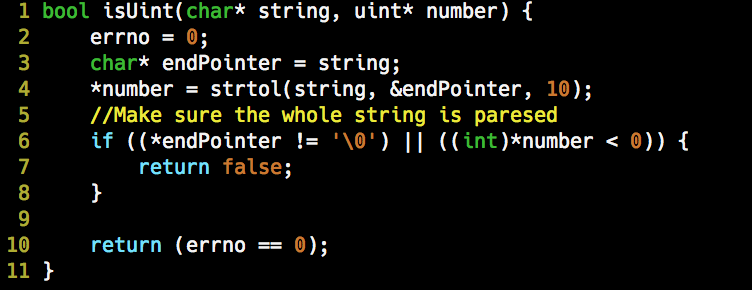
**strncat**(char\* dest, const char\* src,size\_t n);

**strlen**(const char\* s) (Does not include the \0 in size);

**File Opening/Reading (w/ realloc)**



// STRTOL



**Network Functions**

**socket**() = creates comm

**bind**() = attach local address to socket

**listen**() = willing to accept connections (takes queue)

**accept**() = wait for a connection to arrive (blocking)

**connect**() = attempt to establish a connection

**send**()/**write**() = send data over a connection

**recv**()/**read**() = receive data over the connection

**sendto**() = send datagram (UDP)

**recvfrom**() = receive a datagram (UDP)

**close**() = release the connection

**shutdown**() = closes and endpoint of the connection

**htonl**() = converts from uint\_32t host byte order to uint\_32t network byte order.

**htons**() = converts uint\_32t host byte order to uint\_16t network byte order.

**ntohl**() = converts uint\_32t host byte order to uint\_32t host byte order.

**ntohs**() = converts uint\_32t host byte order to uint\_16t hostbyte order.

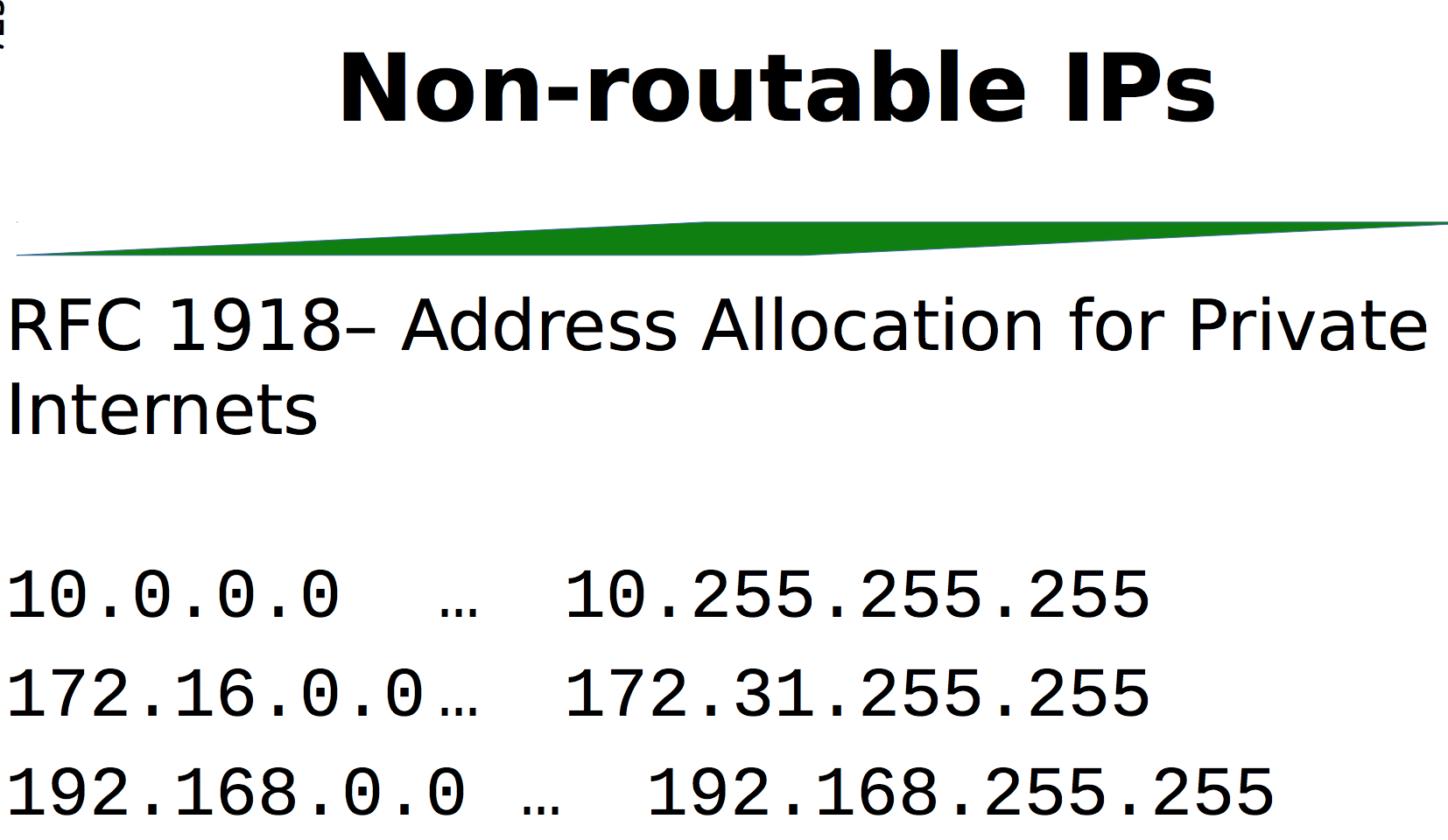
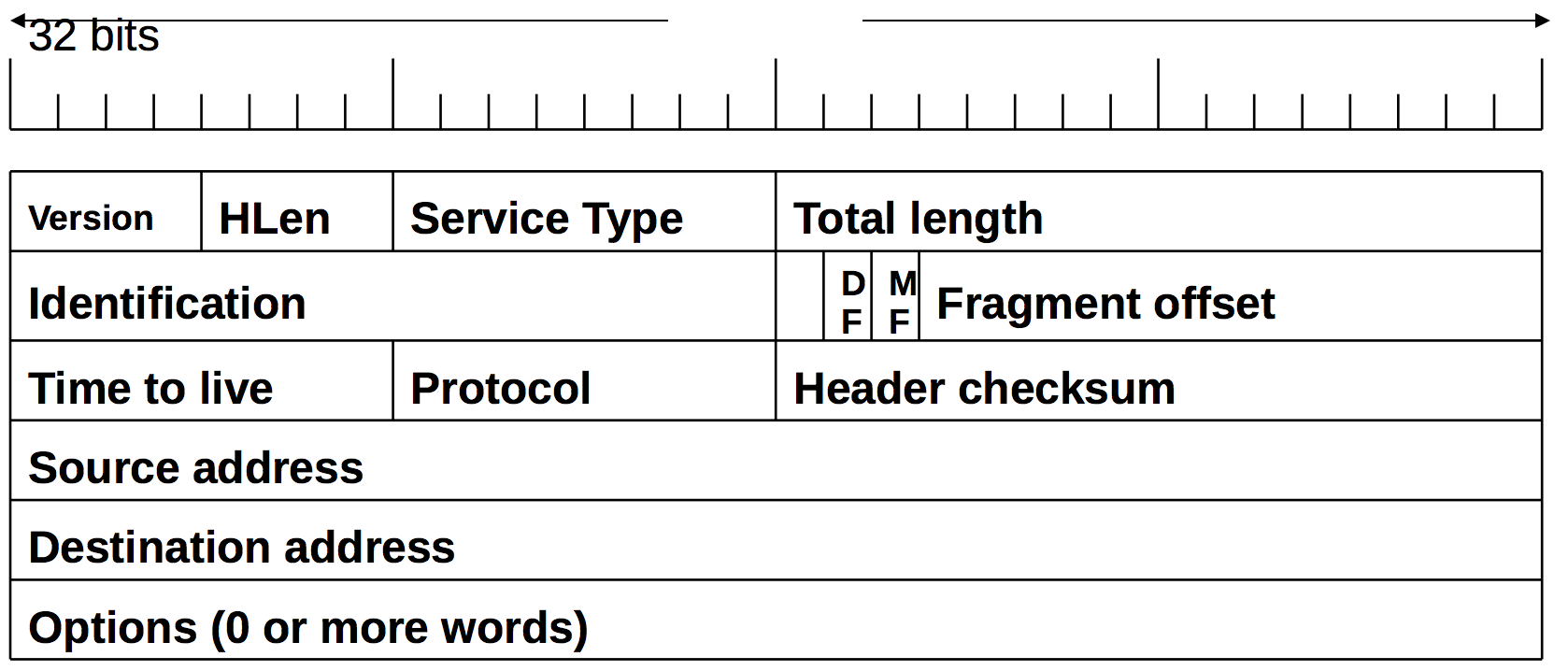
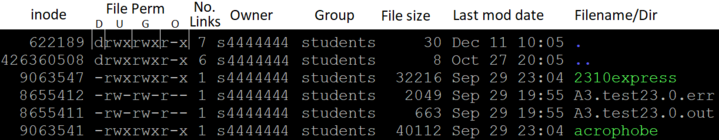
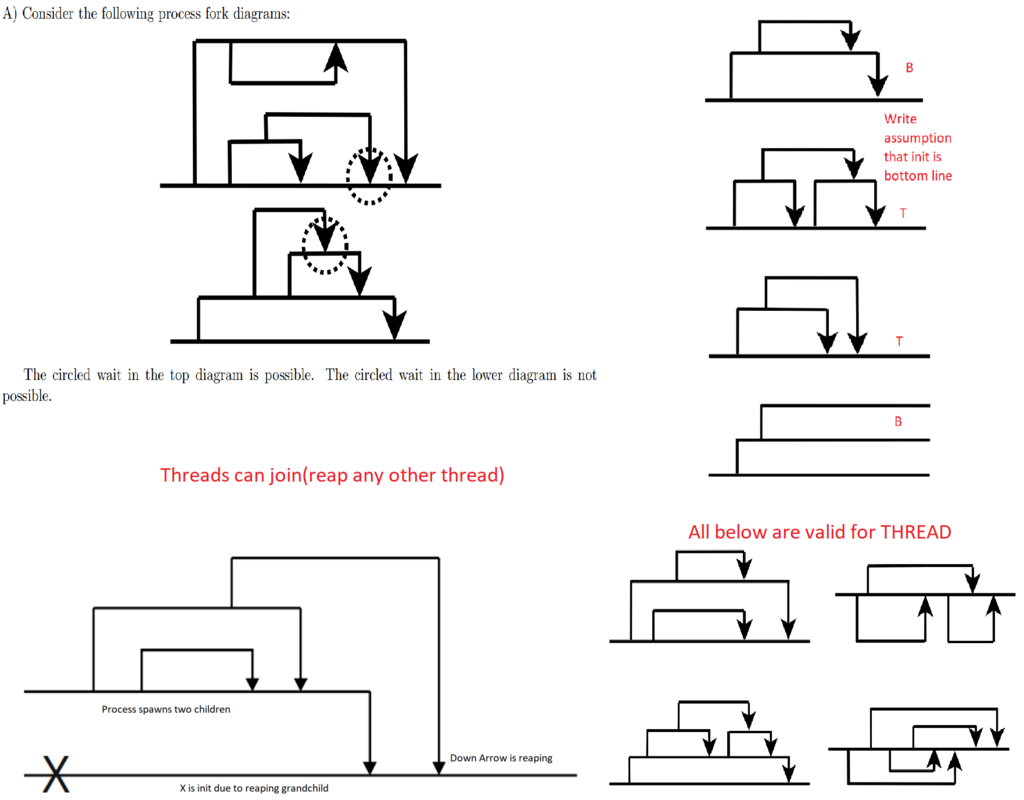
**inet\_aton**() = string to dec, notation (assumes host byte order)

**inet\_ntoa**() = dec. to string notation (assumes host byte order)

**setsockopt**() = close port immediately on program exit (frees for immediate reuse).

**getnameinfo**() = IP address to URL

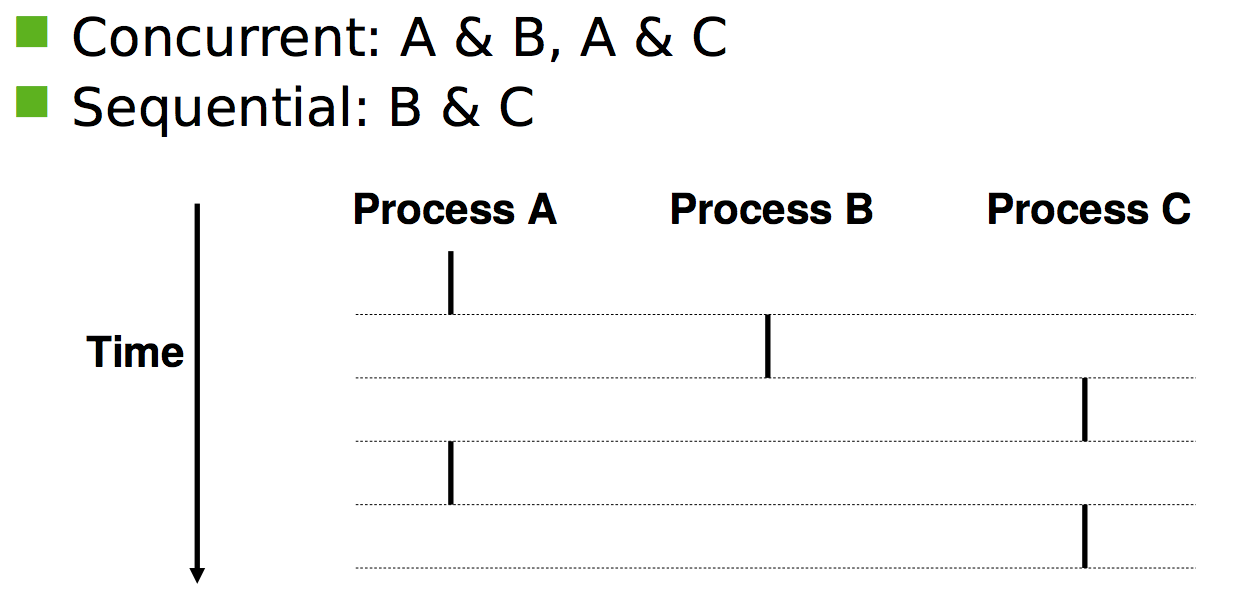




**Multiprocessing**

Process: abstraction of CPU time -> instance of running program.

Attributes: own logical flow/address space/system resources.



**pid\_t pid** = fork()

int fork()

void **exit**() {exits process and flushes all buffers}

**\_exit**(){exits immediately, may not flush buffers}

**atexit**(void (\*func)(void)) {register a callback to run at exit}

**Zombie**: terminated process that has yet to be reaped.

**Reaping**: performed by immediate parent or init.

Int **wait**(int\* child\_status) {suspends current process until a child terminates – child\_status is filled with exit status of child proc}

**waitpid**(pid, &status, options) **if no options set to 0** {like wait for specific pid}

use WIFEXITSTATUS

**Status Macros** (All only matter if non 0)

**WIFEXITED**(status) – (bool) did process end normally

**WEXITSTATUS**(status) - (int) exit status

**WIFSIGNALED**(status) – (bool) ended because of un-caught signal

**WSTOPSIG**(status) – (int) signal that stopped the child

**WTERMSIG**(status) – (int) signal that ended the child

execl functions {final arg is NULL - sentinal}

execlp(“ls”, “ls”, “-la”, NULL);

**Networks**

**Subnet Mask** =

1. From MSB, find first difference. Blocks before = 255, after = 0, block of = set all to 1 until one before MSB.

**CIDR** = like blocks remain, following blocks set to 0. \X, where X = number of bits from MSB to one before first difference.

**Network Address** = Bitwise (of any) & w/ subnet mask.

**Broadcast Address** = Inverse of Subnet Mask + Network Address (by block).

**Interface Count** = (2^(32-x))-2 (where x is CIDR\x)

**Invalid Netmask** = Non Contiguous from MSB Bits (eg 255.255117.0)

**Gateway Address** = closest internal address to the internet? CHECK THIS

Physical > (Data) Link > Network > Transport > Application

MAC > IP > Port > Socket

Copper > Wifi > Router > >

**File Systems**

bytes/1024 = Kib

Kib/(1024)^2 = Gib

Read Time

(If already cached take 1 (min of 1))

direct = 1 | indirect = 2 | dbl. indirect = 3

(note the 0 or 1 starting position (eg 10 direct is [0,9]))

[direct][in-direct1][in-direct2][dbl.]..

Regex

^ Start Anchor

$ End of String

? Single any val

a? Zero or one of ‘a’

a+ One or more of ‘a’

a{3,} 3 or more of ‘a’

a{3,6} between 3 and 6 of ‘a’

\s any whitespace

\* Any characters, any length

Print Question Pitfalls (Why Dr Joel, why?!?!?)

* Float values
* Semi colons
* Switch (breaks)
* Missing Braces
* Pointer arithmetic

int (\*functionPtr)(int, int);

A pointer to a function which takes an array of function pointers (each taking and returning a string) and returns one of the function pointers it was passed

char\* (\*(\*foo)(char\* (\*[])(char\*)))(char\*);

struct StructThing {

int elem1;

char elem2;

}

typedef struct {

int elem1;

char elem2;

} NewType;

typedef int foo; (another name for int)

**Linux commands**

**cat** - looks at contents

**cd**

**chmod** [ugoa] [+-] [rwx] - u = user/g = group/o = other/a = all

**chgrp** [group] [file] - change the owner group of the file - has recursive

**chown** [owner] [file] - change the ownder of the file - has recursive

**cp** [-r (recursive)] [source] [destination] (can copy files or dirs)

**cut** -d[delim] –f[columnum] file (e.g. cut –d, -f1 foobar --- gets the first column, comma sep of foobar).

**du** [-h (human redable) s (summary instead of each of the individual)] - disk usage (file size)

**echo** [value]

**exit**

**export** PATH=$PATH:~/DIR

**fg** (brings suspended programs to the foreground) [process number to bring fowards]

**find** [start][-name [name]] - accepts wild cards

**grep** [input] [file [r (recursive)] [-l (only return file)] [-c (count of results)] - returns lines from file with input (name can use wildcards)

**head** [-n[-]K (Show the first K lines or if negative all bar the last K lines)]

**hostname**

**kill** [-9 forces it to die without clean up][pid] - kill running processes

**ln** [-s] [target] [link\_name]

**ls** [-a (all)][-d (just this dir)][-l (show more details)][-i (show inodes)] - can use wildcards eg (\*.pdf) OR ??.pdf (pdf with two letter names)

Sort by -U none, -S size, -t time, -v version,-X extension.

**man** [-k (search for man pages with the input word)][num - open to that page]

**mkdir**

**mv** (move)

**ps** - [u[$USER] (processes of user) f (full info)] list running processes

**pwd**

**rm** [-r recursive][-f force]

**rmdir** (fails if not empty)

**sort** [-d sorts to alphabetical][-n sort by string numeric values]

**tail** [-n[-]K (see head)]

**tar** [c (create) v (verbose) f (file we want to act on) z (zip/compress) x (extract)]

**uniq** (COMPARES ADJACENT LINES) [-d (print duplicates) –c (prefix line with count of occurences) –i (ignore case) –u (print unique lines) –sN (skip first N chars) –wN (only count first N chars))

**wc** [l (lines)] - word count

Special

"| less" - show only one page of details at a time

& - postfix runs command in background

&& - runs first command before second

`` - gets value of expressions before running rest of command

Paths

"." current directory

".." directory up

"~" home directory

Commands

^C - stop currently running program

^Z - suspend currently running program

^S - locks up terminal?

^Q - fixes ^S

Redirection and pipes

> - redirect stdout and overwrite

>> - redirect stdout and append

2> and 2>> - redirect stderr instead

< input instead of using keyboard

| - send output to following program

**Virtual Memory**

**Offset** = Virtual Address % Page Size

**Page** = Floor(Virtual Address / Page Size)

**Frame** = Floor(Physical Address / Page Size)

**Physical Address** = (Frame \* Page Size) + Offset

**Virtual Address** = (Page \* Page Size) + Offset

**TLB** = Translation Lookaside Buffer (cache for translation of VM to PM).

**TLB** – Pages accessed to read VAs

1. Can we use cache? If so, + 0.
2. Follow L1, L2 & PA = + 2.
3. Page->Frame mapping will now be in cache.

Page Tables between processes: Pages change, frames stay the same.