

java -jar /home/kurmasz/public/bin/JLS.jar

Gu2SvHI

Wget "" pastebin

-h help

| tee filename.txt output to std and file

Find \n\n replace \n

Dirbuster and gobuster for finding /dev ... use on <https://cis.gvsu.edu/~scripps/cs162/>

Vim .bashrc

export PATH=\$PATH:/home/kurmasz/public/bin

. .bashrc

Cd change directory

Ls list

Pwd present working directory

~ is home directory just cd goes back home too

Cd ~/cs241

Cd ~muldecha

Touch creates an empty file

Touch testfile

Touch test.txt

Rm -r ~/* removes all files in home directory

Rm remove

Rm test.txt

Rm -l

Rm -l test.txt prompts before removal

Rm -l * ask to remove all of my files

Ls * is specific search

Ls *.txt gives all txt files

. in front of file makes it hidden

? for single character

Ls test?.txt

Ls – l for additional info

Chmod u+x test.txt makes the file executable

Chmod ugo+x test.txt changes for user group and other

Ssh muldecha@eos22.cis.gvsu.edu do this on command prompt to use cmd

Chmod go-r test.txt removes other users from reading the file

Use up arrow for previous command

Double tab shows auto fill options

Cd – goes back to the last place you were

:wq exits vim and saves

:w saves

Mv alice.txt

Vim alice.txt

Dd deletes a line

Dw deletes words

D\$ delete to the end of the line

D^ deletes to the beginning of the line

Db deletes previous word

^ or \$ to get to beginning or end of line

W goes forward

B goes backward

3 up arrow goes up 3 lines

Y is copy

Yy is copy whole line

P is paste

10yy copies 10 lines

V is visual and lets you use cursor for copying

Vimtutor is the tutorial

Vim hello.c

```
#include <stdio.h>
```

```
Int main(){
```

```
Printf("hello world\n");
```

```
Return 0;
```

```
}
```

Esc :colorscheme slate is cool

Clang hello.c compiles programs to a.out

Clang hello.c -o hello names it hello

./hello.c runs program

In vim /Alice searches for words

Vim ~/.vimrc reads on startup

Echo \$PATH shows path

Ls -d /home/muldecha/bin shows home bin

Mv hello /home/muldecha/bin/

Hello runs program

/home/muldecha/cis241s21/hello runs program as well

Which tells me where a command is

Which hello tells the path

Type hello shows path and if its hashed

Vim ~/bin/no creates a file called no

#!/usr/bin/bash makes a shell program

Echo "no" prints no

Chmod u+x ~/bin/no makes it executable

No runs the program

Ls -l ~/bin lists bin

Cp ~/bin/no ~/bin/man copies no and renames it man

/usr/bin/man ls runs the program also

Hash -r deletes hash records

If a thing is hashed then it will run the right way

It'll run the hashed version

Alias ls='ls --color=auto'

Alias ls = list

Github.com/new

Private

Cd gitdemo

git remote add origin <https://github.com/ChaseMulder/hello1.git>

git branch -M main

git branch tells what branch we're on

git push -u origin main

ChaseMulder

gorillazCool1

ctrl c to exit out of vim

ctrl c for a new command line to ctrl a ctrl x things

```
printf("you entered %d\n",i);
```

```
%d
```

```
%x
```

```
%o
```

```
%9d
```

%09d use 9 characters to print the number with leading zeros

```
%+9d
```

```
%9.5d
```

```
%.5d
```

```
%.*d\n ,7,2,i
```

Doubles

%lf &f

%.3lf

%lf

%lg

Man -a printf

Auto indent is in normal mode(not insert) gg=G

#include <math.h>

Clang -lm ctesting.c

|| logically

| bitwise

^ xor

~ inverter / not

<< bit shift >>

>> right shift 1001 becomes 0010

<< left shift

Int32_t %PRId32

Short %hd

Long %ld

Long long %ll

Unsigned int %u

Unsigned long %lu

%hu unsigned short

Float

Double

Long double

```

#include <float.h>

FLT_MAX DBL_MAX LDBL_MAX

Sinf sin sinl

%.20f %lf %Lf &

Clang -lm name -lm if using the math library

#include <tgmath.h>

Sin only with tg instead sinf

%c to print chars

#include <stdbool.h>

Printf("%s\n",x?"true":"false");

#include <complex.h> for complex numbers

Double complex x = csqrt(-2);

X = sqrt(-2.0+0I);

Printf("x is : %lf+%lfi\n", creal(x), cimag(x) );

Switch(num){

Case 9:

Num = 'A';

Case 8:

Num = 'B';

}

Scanf("%s\n",s);

If(strcmp(entry,"quit"));

%*d ignore numbers

%*c ignore characters

%*lf ignores floats

```

Echo \$? Return value of the last program run

V visual mode

Esc normal mode

U in normal more for undo

Ctrl + r redo

Va_start

Va_end

How to change an environment variable path ps1

Ps1 is to modify the command prompt

How to use git push, pull, init, checkout

Preprocessor statements vs c statements #if vs if() # is being run at compile time, so the condition is being checked when the code is compiled – it's a one time decision vs a regular if which tests when the program is running

Why shouldn't you alias cp to be cp -l cp -l asks to overwrite files, don't get accustomed to that behavior

Difference between macro and function – recursion is not possible with macros

Macros don't need to be passed variables with types – function do

Macros are more efficient than function because they don't involve function calls

How to make a permanent alias – put it in your bashrc or bash profile – for all put it in /etc/all profile

Bashrc is run everynew shell

Bash is run once

Export PATH=~/.cis241/bin:\$PATH beginning of path

Export PATH=\$PATH~/.cis241 end of path

#include *dice.c*

Clang diceroll.c

Clang -c compiles but doesn't link files

Ls -lh > head

| is not or, it forwards on command to the next

Head -15 gives first 15

& not proper && proper

| bit wise || or

Anything not 0 is true

Know chmod

-rw-r—r-- which of the following things can a person who is not a user do?

Dash then first 3 owner, 2nd 3 group, last 3 are other/everyone else

Dash is a file d is a directory

U for user g for group o for other

Chmod u-r ~/filename

R w x for read write execute

#define something(x)\ use slashes instead ;

```
#define square(x)/ x*X
```

```
Int square(int x){
```

```
  X*x}
```

Square(5+3.3); neither run because macros

Macros are doing text substitution and are not evaluating the 5+3.3 argument

!350 runs command 350 from history

!! runs most recent command from the history

%x

%99s limits string to 99 characters

Scanning strings no &

There's no way to ask an array for its size... no sizeof()

Double is more precise than float and uses more bits to store it

UInt64_t we know this size

Long we don't know this size

Make CC = clang \$(CC)

Command line stuff overrides stuff in the make file

- Bash scripts must start with `#!/usr/bin/bash`

(lldb) breakpoint set -name rolldice breaks when it runs into that function

- C concepts
 - Overflow
 - What happens when you add 1 to the max integer?
 - You get a negative number
 - If you compile a program that does this explicitly you get a compiler warning
 - But the compiler can not detect it if the overflow is happening based on user input
 - No runtime errors or warnings happen in this case either, you simply get wrong output

- C library functions
 - rand()
 - from stdlib.h
 - returns random number
 - might not be a good random number generator, especially in older implementations
 - Must seed the generator first
 - Use `srand(unsigned int)` to seed the generator
 - A common way to get the seed to pass to `srand` is with `time(NULL)`
 - Need `time.h` for `time(NULL)`
 - results based on current time, so seed is likely to be unique each time program is run
 - For testing programs, sometimes a fixed integer is used
 - This provides consistent results across tests

• Linux concepts

○ environment variables

■ `$?`

- contains return value of previous command

lldb is the debugger

`x[i] *(x+i)`

`-du size`

`Du -s . size of directory`

Call `stat` on current directory

Sort array by size

Ls -S | head n-5

Anything node / sorting on exam

Linked list examples find the errors in code***** not setting end thing in structure to point to null –

LL intlist.c week 9

Wrong: typedef struct{ error is declaration before use – creating list pointer before it knows what list is

```
    Int x;                                correct it by struct list{ struct list *y;
    List *y;
} list;
```

```
Struct list *l = malloc(sizeof(struct list));
```

```
l->y = NULL;
```

```
x = $(du -s $dir)    x = size of the directory
```

```
y = $(du -s $file)
```

```
put = $(echo ($file/$dir)*100) | bc)
```

In context of the weekly content what week is most important

Week 9 shell script answer

Find the errors in a shell script

-doing floating point math without using bc ex. \$(echo (\$file/\$dir)*100) | bc)

Wrong: let z = \$x + \$y

Wrong: \$((z=\$x+\$y))

Factorial.sh shell script with functions

Filemanage.sh select case

Mc on signals

Correct syntax for function pointers bubblesort.c cmpfunc compare

Without typedef: //void bubblesort(int a[], int n, int (*compare)(int, int))

Mc sed

Short answer comparison : would it be better to do a thing in awk or sed

Should we use a c script or a bash script for a thing

Errors in a bash script: contains #!/usr/bin/bash

What was the thing that had specific spacing and [] vs [[]] – either one can be used

Common errors in malloc

Bash-> 0 is true everything else is false

[[\$x<\$y]]

Echo \$?

Bash know when to use the \$ sign

Let x=5+6 let cannot have spaces x=5+ 6 wrong

((x = \$y + z)) doesn't work z needs \$

Struct list* x = malloc(sizeof(struct list*)) wrong

Struct list* x = malloc(sizeof(struct list)) right

Grep searches : grep "Alice." alice.txt finds alice, because . is wild card

```
Int *z;
```

```
Scanf("%d", z); wrong
```

```
Int y;
```

```
Int *z = &y; or int *z = malloc(sizeof(int))
```

```
Scanf("%d", z); right
```

```
Getc(stdin)
```