ECE220 Lab1

Introduction

TA: Matt Potok (Poe tock)

Email: potok2@illinois.edu

Office Hours:

- Monday – 6:00 PM to 8:00 PM
- Thursday – 6:00 PM to 8:00 PM

Course websites:

- Course Wiki: https://wiki.illinois.edu/wiki/display/ece220su2/ECE+220+SU18+Home+Page
- Course Github: https://github-dev.cs.illinois.edu/ECE220SU18/_release
- Piazza: https://piazza.com/class/jexl8le0xnb3zz

Class Overview

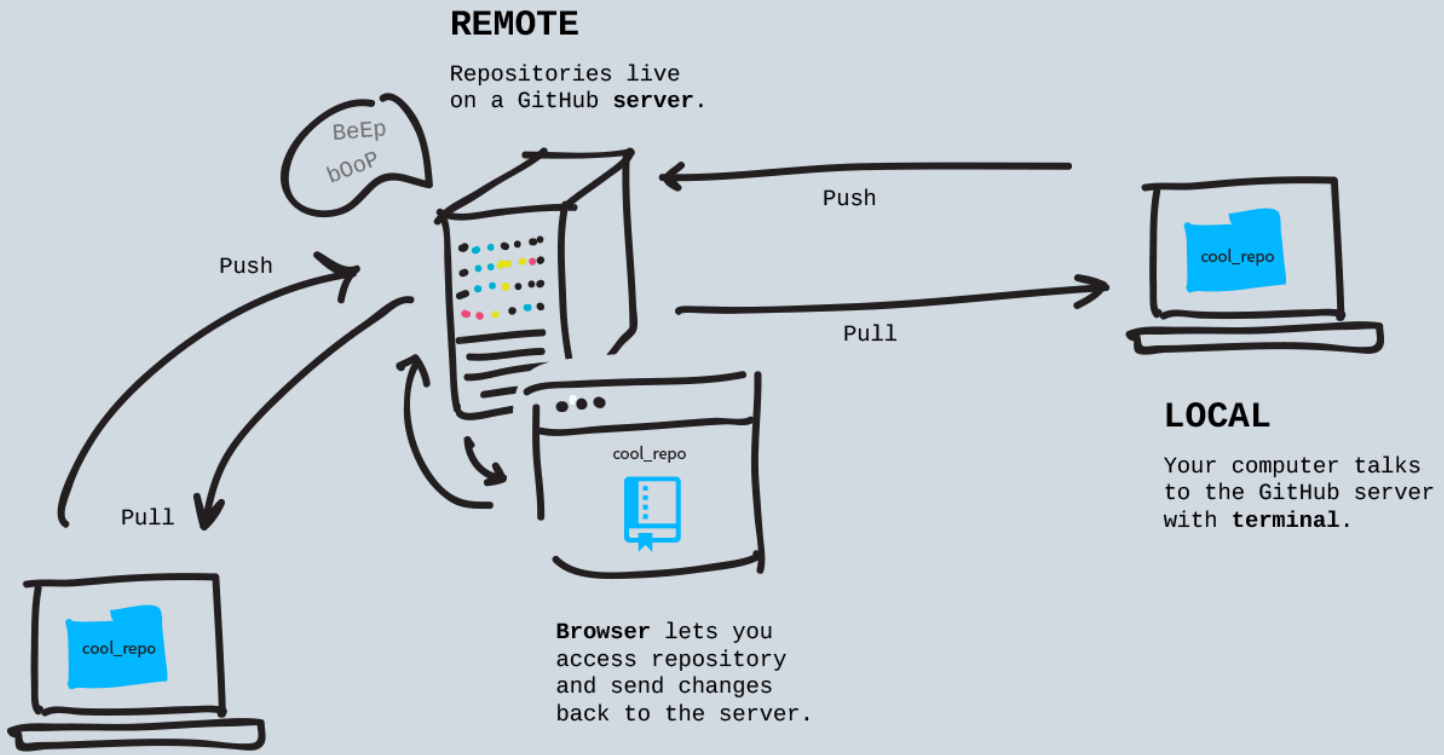
Machine Problems:
- Available on **Monday** before the lab
- Due next **Monday 10:00 PM CST**

Labs
- Weekly on **Mondays 3:00 PM to 4:00 PM CST**
- Optional lab assignment
  - 10 extra points per assignment
  - Available on **Monday** before the lab
  - Due next **Monday 10:00 PM CST**

Quizzes
- Weekly on **Thursday 8:00 PM to 9:00 PM CST**

Git/Github
- Version control system
- Retrieve and submit assignments

Git Overview



Setting up Git and Lab1

Login at https://github-dev.cs.illinois.edu/

Follow instructions at https://github-dev.cs.illinois.edu/ECE220SU18/_release

Lab1a – Hello World!

Two assignments:

- lab1a – Hello World!
- lab1b – Printing a hexadecimal number

More information about the lab at https://wiki.illinois.edu/wiki/display/ece220su2/Lab1

Complete lab1a and see ‘Hello <your name>!’ printed to the console

Commit changes to your local repository.

Push changes to the remote repository.

Working Remotely

Personal computer

- Install LC3 tools: https://wiki.illinois.edu/wiki/display/ece220su2/Installing+LC-3+tools+on+your+machine

EWS machines

- SSH/X-forwarding
- Remote desktop environment with FastX
- Load LC3 tools with *module load lc3tools*
- Instructions at https://wiki.illinois.edu/wiki/display/ece220su2/Working+Remotely

SSH Keys:

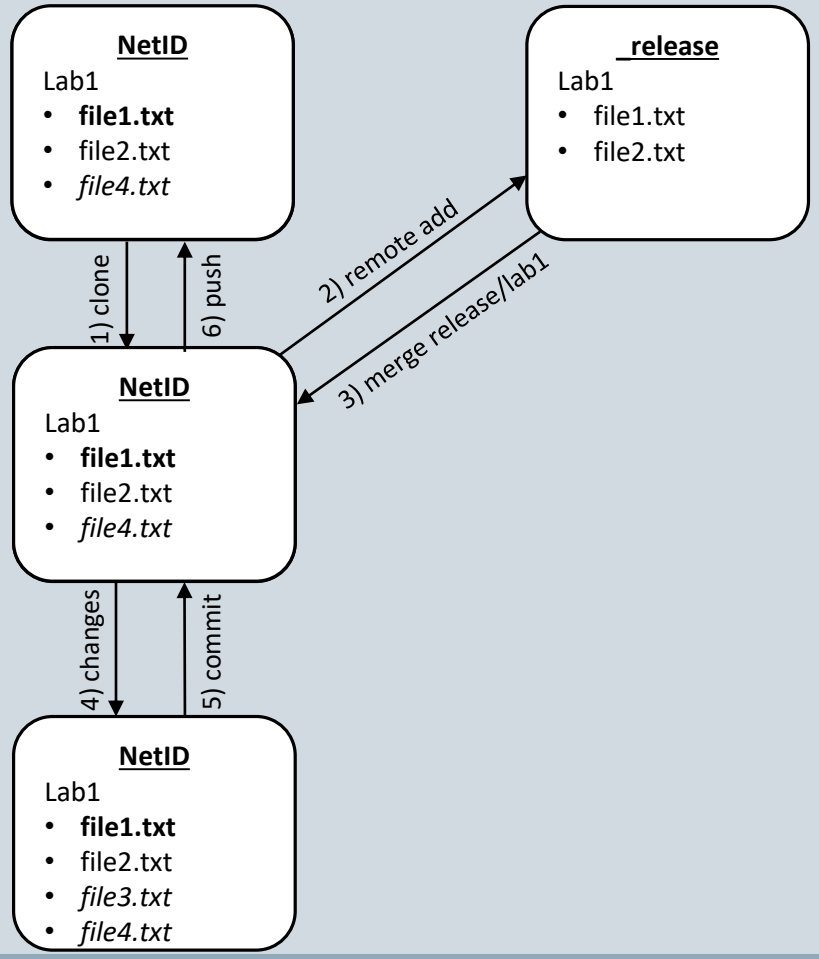
- Follow instructions at https://wiki.illinois.edu/wiki/display/ece220su2/Setting+up+SSH+Keys

Git Review

Remote Repository

Local Repository

Workspace



git status @ 3)
- unmodified tracked
  - lab1/file1.txt
  - lab1/file2.txt

---

git status @ 4)
- unmodified tracked
  - lab1/file2.txt
- modified tracked
  - lab1/file1.txt
- untracked
  - lab1/file3.txt
  - lab1/file4.txt

git status @ 4.5)
- unmodified tracked
  - lab1/file2.txt
- untracked
  - lab1/file3.txt
- changes to commit
  - lab1/file1.txt
  - lab1/file4.txt

---

git status @ 5)
- unmodified tracked
  - lab1/file1.txt
  - lab1/file2.txt
  - lab1/file4.txt
- untracked
  - lab1/file3.txt

MP1 and Lab1b – Printing a hexadecimal number

This is a test of the counting frequency code.
AbCd...WxYz.

@ 000F

A 0002

B 0001

…

Y 0002

Z 0001

MP1

- Given a histogram of letters and symbols, print it out to the console.
- More information at https://wiki.illinois.edu/wiki/display/ece220su2/MP1+-+Printing+a+Histogram

Lab1b – Printing a hexadecimal number

What LC3 ASM instruction can we use for printing?

- LC3 ASM instruction OUT – write character from R0[7:0] to console

Suppose given R3 with xCA75 (x1100 1010 0111 0101)

What order do we want to print hexadecimal characters to the console?

- C, A, 7, 5 (left to right)

How to move each hexadecimal character into R0?
x1100 1010 0111 0101 -> x0000 0000 0000 1100

- Check MSB of R3 to see if a 1 or 0 where 1 indicates R3 is negative and 0 indicates R3 is either zero or positive
- Add 1 to R0 if R3 is negative
- Left shift R3 and R0 by 1

Lab1b – Printing a hexadecimal number

Whole process from previous slide

Initially have R3 with x1100 1010 0111 0101 and R0 with x0000 0000 0000 0000

1. x1001 0100 1110 1010 and x0000 0000 0000 0001
2. x0010 1001 1101 0100 and x0000 0000 0000 0011
3. x0101 0011 1010 1000 and x0000 0000 0000 0110
4. x1010 0111 0101 0000 and x0000 0000 0000 1100

Are we done yet?

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Lab1b – Printing a hexadecimal number

Now have R0 with x0000 0000 0000 1100

What happens if we call OUT now?

- Tries to print xC to the console which is the character for a new page.

How can we correctly print 'C' to the console given x1100?

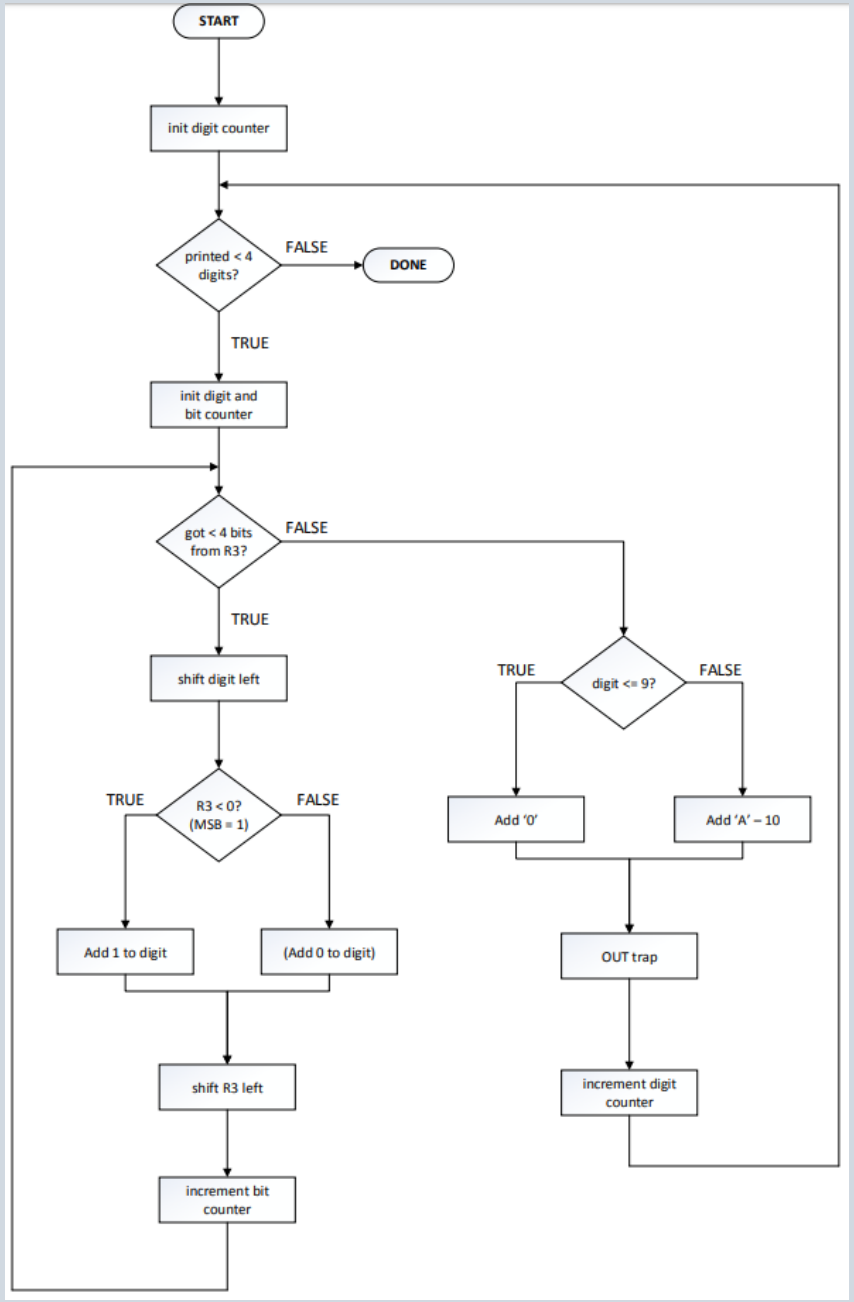
- Need to convert x1100 to ASCII letter 'C' which is x43 (x0100 0011)

Do we need to check the number in R0 before adding any offsets?

- Yes!

Which offsets do we need to add?

- x30 for 0 – 9 and x41 for A – F



Lab1b Flowchart