CS 354 - Machine Organization & Programming
Tuesday Jan 23, and Thursday, Jan 25, 2024

Week 1 Objectives (at a minimum, student should be able to)

- use ssh to connect to their CSL account
- use cp to copy files (e.g. .vimrc from /p/course/cs354-deppeler/public/ to ~/.vimrc)
- use scp to copy a file from your CSL account to your local computer
- use scp to copy a file from your local computer to your CSL account
- use vim to create and edit a C program source code file
- use gcc to build a Linux executable "program" from a C source file
- run a program that was built from C source code file(s)
- use gdb to step through program and examine variable values
- learn and use other Linux C dev tools (commands) as needed
- learn basic C structure and logical control flow statements

This week

Welcome	Basic C Programming on Linux
Course Intro Syllabus Canvas Web Pages Exams Projects Quizzes Activities	C Logical Control Flow C Program Structure Remote Connect to CSL Account Coding in C Remotely Edit your Source Compile Run/Debug/ Submit

NextWeek

Topics: Finish C Program Structure and Control, Variables & Pointers

Review:

K&R Ch. 2: Types, Operators, and Expressions
variable names, data types, constants, declarations
arithmetic/relational/logical operators, assignment, precedence

K&R Ch. 3: Control Flow
statements & blocks, if-else & else-if, switch, while, for, do-while

K&R Ch. 4: Functions & Program Structure
basics, parameters, return values, scope rules

Do: read course "Information and Policies" pages linked to course website
access CS Linux lab computers, try Linux commands and tools (vim, gcc, gdb, man)
check out course Piazza site

C Logical Control Flow

```
int main(int args, char *argv[]) {
…;
return 0;
}
```

Sequential

executionstarts in main(), flows top to bottom, does one statement after another

Selection

if, if else, switch

if (42) evaluates to true

→ Which value(s) means true? `true` (42) (-17) 0 ‘0’ NULL ‘\0’

True

? depends on what it is defined as

False

False 0x0

False

Want the 0 value not character 0

→ What is output by this code when `money` is 11, -11, 0 ?

assigns 0 to money, not equality

```
if (money = 0)       printf("you're broke\n");
else if (money < 0) printf("you're in debt\n");
else                 printf("you've got money\n");
```

in C
- when money is 11, output is “you’ve got money”
- when money is -11, output is “you’re in money”
- when money is 0, output is “you’re money”

if equality ==
- when money is 11, output is “you’ve got money”
- when money is -11, output is “you’re in debt”
- when money is 0, output is “you’re broke”

→ What is output by this code when the date is 10/31?

```
if ( month) == 10
   if (day) == 31
      printf("Happy Halloween!\n");
else
   printf("It's not October.\n");
```

dangling else, match with nearest if and not by indentation

use {} to control if-else

M	D	Result (expected)	Result (actual)
10	31	“Happy Halloween”	“Happy Halloween”
1	12	“Not October”	— (no output)
12	32	? unsure expected	“Not October”
9	31	“Not October”	— (no output)

switch cant use CHAR STRING

Repetition

```
int k = 0;
do {
   printf("%i\n", k);
   k++;
} while (k < 11);
```

\n, newline and flush buffer

```
int i =0;
while (i < 11) {
      printf("%i\n", i);
      i++;
   }
```

format

post increment

```
for (int j = 0; j < 11; j++) {
   printf("%i\n", j);
}
```

doesnt matter if pre or post increment

C Program Structure

❋ *Variables and functions must be declared before they're used.*

➢ What is output by the following code?

%d : decimal
%i : integer

```
#include <stdio.h>
```
printf is function in stdio library #printf

```
int bing(int x) {
   x = x + 3;
   printf("bing %d\n", x);
   return x - 1;
}

int bang(int x) {
   x = x + 2;
   x = bing(x);
   printf("BanG %d\n", x);
   return x - 2;
}

int main(void) {
   int x = 1;
   bang(x);
   printf("BOOM %d\n", x);

   return 0;
}
```

bing 6
BanG 5
BOOM 1

bing	x 6	and returns 5 to bang
bang	x 1 -> 3 -> 5	and returns 3 to main
main	x 1	

only pass a copy of 1 (x) into bang

Functions

function: like a method - not linked to instance or class

caller function: starts new function

callee function: function being started

Functions Sharing Data

argument: data (values) passed to a function

parameter: variable (location) that stores that value

pass-by-value (passing in): copy of argument's value that is passed to parameter location

return-by-value (passing out): copy of return value passed out of function

return-by-value (passing out):**Remote Connect to your CSL Account**

❋ *Use your CSL Linux account and presented tools for all CS 354 programming.*

1. Connect remotely to any CSL Linux Workstation (login to CSL from your laptop)

a. open your computer's **terminal** application
b. use ssh to secure connect to a Linux network workstation

<shell-prompt>:~$ ssh eyy@best-linux.cs.wisc.edu

shell-prompt: usually user@machine name
(508) deppeler@vm-instunix-04:~$

cslogin: your username for CSL workstations. https://apps.cs.wisc.edu/accountapp/
machine: a physical or virtual machine on the CSL network
emperor-01 ... emperor-07
rockhopper-01 ... rockhopper-09
royal-01 ... royal-30 1366 Comp Sci
snares-01 ... snares-10
vm-instunix-01 ... vm-instunix-99 virtual machines
network: the CSL's network is **cs.wisc.edu**

c. **ssh** eyy **@best-linux.cs.wisc.edu**

Create ~/private/cs354 directory mkdir -P ~/private/cs354/p0

Change to your newly created directory

Create a new directory named projects

Change to projects directory

Print Working Directory

EDIT -- Create your C source code file text-only

1. Create new or open existing file in a text-only editor vim, emacs, nano, pico

```
$vim prog1.c
$vimtutor
```

Why vim? keyboard shortcuts

```
/* File:   input_echo.c
 * Author: Deb Deppeler
 * Desc:   Store and echo the first N characters of user's input.
 * Note:   The newline char \n is replaced by null char \0
 */
#include <stdio.h>          printf, fgets
#include <stdlib.h>         malloc
#include <string.h>         strlen

int N = 8;          global variable

          # of command line arguments
int main( int argc, char *argv[] )
                        array of ptr to character
   // Create space to save string of characters

      char *        input_string = malloc(N) — N bytes                    8 bits in a byte so 8 bytes, so 8 characters

                              return void/generic pointer but autocast to char
   // INPUT: prompt user for input
   printf("Enter a string of chracters: ");

   // INPUT: read keyboard input into input_string variable
   if ( fgets(input_string, N, stdin) == NULL )
      fprintf(stderr, "Error reading %i characters of user input.\n", N);
                                          prints only when fgets is NULL — usually a hardware error
   // PROCESS: Replace '\n' with '\0'
   int len = strlen(input_string);          printf("len=%d\n",len);
   if ( '\n'==input_string[len - 1] ) {
      input_string[len - 1] = '\0';
      printf("replaced \'\\n\' char at index %i with \'\\0\' \n", len-1);
   }

   // OUTPUT: print CS login to terminal
   printf("First %d chars of your input string: %s\n",len,input_string);

   // RETURN
return 0;
}
```

COMPILE, RUN, DEBUG, SUBMIT

2. Compile -- build executable from C source

```
$gcc prog1.c
```
gnu c compiler src

```
$gcc prog1.c -Wall -m32 -std=gnu99 -o prog1
```
-Wall generate all warnings
- -m32 use x32 ABI application binary interface in Linux (x86-64 with 32 bit pointers)
- -std=gnu99 select c dialect like java for loops
- -o prog1 give output a specific name

3. Run -- run executable (program) from command line

```
$./a.out
```
./ for current directory

→ Why a.out?

assembler output

```
$./prog1
```

4. Debug

1. Add print stmts: printf("strlen=%i\n", strlen(input_string));
2. Use gdb add -g option to gcc command

Write test harnesses create test cases

5. Submit work to Canvas assignment (required if working from personal computer)

- DOWNLOAD copy from CSL to current directory on your local machine

scp CSLOGIN@best-linux.cs.wisc.edu:/home/CSLOGIN/private/cs354/hello.c .

location to copy to, current directory

- Hard-Refresh Canvas assignment page
- Upload files from your local machine

If file upload does not complete, the page is "stale" or you have missed late due date.
Close ALL browser windows and re-login to Canvas and refresh your assignment.

Try some Linux File System Commands

command shell

→ How do you?

list the contents of a directory? ls

show details of each file? ls -l
show hidden files in the directory? ls -a

get more information about commands? man ls

display what directory you're currently in? pwd

copy a file? cp scp secure copy -r recursively copy

remove a file? rm

move to another directory? mv

move "up" a directory? cd ..

make a new directory? mkdir

remove a directory? rmdir

rename a file or directory? mv