CS 354 - Machine Organization & Programming
Thursday, September 7, 2023

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

Today

Basic C Programming on Linux	
C Logical Control Flow C Program Structure Remote Connect to CSL Account Coding in C Remotely Edit your Source Compile Run/Debug/ Submit	Try more Linux commands Next Week: Pointers and 1D arrays

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

Copyright © 2021-2023 Jim Skrentny

C Logical Control Flow

Sequential

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

function - stand alone

Selection conditional

expression

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

true	42	-17	0	'0'	NULL	'\0'
?	T	T	F	T	F	F
not reserved in C						bit pattern null character

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

0 is false   assignment

```
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");
```

11: got

-11: got

0: got

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

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

dangling else matched with nearest if

M	D	Result
10	31	T
10	30	
11	31	
13	31	
0		

switch like java, no string

Repetition

✯

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

closest to assembly code

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

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

post
j++;

y=j++;

inc after
use value

pre
++j;

y= ++j;

inc before
use value

C Program Structure

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

➢ What is output by the following code?

prog.c

```
#include <stdio.h>
                standard io
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;
}
                 reserved word
int main(void) {
   int x = 1;
   bang(x);
   printf("BOOM %d\n", x);

   return 0;      program
                  ran correctly
}
```

- NOT methods, functions
- main relys on bang

→ bang is def before main

Functions

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

caller function: starts a new function executing

callee function: the fnc being started

Functions Sharing Data

argument: data / "value" being passed

parameter: variable / "location" that stores the data

pass-by-value (passing in): copy of arg passed in

return by value : copy of value being returned

Copyright © 2021-2023 Jim Skrentny

~~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>:~$
```

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
snares-01 ... snares-10
vm-instunix-01 ... vm-instunix-99
network: the CSL's network is **cs.wisc.edu**

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

home dir
↓
Create ~/private/cs354 directory
mkdir ↑
only usr can access

Change to your newly created directory
cd ~/private/cs 354

Create a new directory named projects
mkdir projects

Change to projects directory
cd projects

Print Working Directory
pwd

Copyright © 2021-2023 Jim Skrentny

EDIT -- Create your C source code file

1. Create new or open existing file in a text-only editor

```
$vim prog1.c
$vimtutor  ← easy vim cmds
```

Why vim?

```
/* 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>
#include <stdlib.h>
#include <string.h>

int N = 8;

int main( int argc, char *argv[] )

   // Create space to save string of characters
   char * input_string = malloc (N);

   // 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);

                           usr typed msg ends with \n
   // 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;
}
```

type it

Copyright © 2021-2023 Jim Skrentny

COMPILE, RUN, DEBUG, SUBMIT

2. Compile -- build executable from C source

```
$gcc prog1.c
```

CLAs

command

```
$gcc prog1.c -Wall -m32 -std=gnu99 -o prog1
```

-g

rename output

-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
```

→ Why a.out? "assembler output"

current directory

```
$./prog1
```

4. Debug

start

next - over

step - in

b main - break point main

p varname - print value

☆ 1. Add print stmts:
2. Use gdb

Write test harnesses

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 .

curr. direct.

- 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

Copyright © 2021-2023 Jim Skrentny

command shell

→ How do you?

list the contents of a directory? ls

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

get more information about commands?

display what directory you're currently in?

copy a file? cp ___ ___

remove a file? rm ___

move to another directory? mv ___ ___

move "up" a directory? cd ..

make a new directory? mkdir ___

remove a directory? rm dir

rename a file or directory? mv ___
name

Copyright © 2021-2023 Jim Skrentny