CS101 Computer Programming and Utilization

Milind Sohoni

June 6, 2006

Course Organization

Lectures and Laboratory

- Two classes per week, each of 1 hour.
- 2 hours of laboratory work.

Resources

- Linux Handbook.
- gcc handbook.
- Tutorial and Worksheets.

Machines

Machines are devices which *execute* or *implement* a given operation. Machines have instructions and states.

Example

Stapler:

Current State	Instruction	Next State	Action
Ready	Hit	Ready	staples the sheets

Example

Vending Machine:

Current State	Instruction	Next State	Action
Ready	Tea	Ready	Vend Tea
Ready	Coffee	Ready	Vend Coffee

A more interesting example...

The state is a 2-tuple, such as [off,on], saying that electricity is off while water is on.

Water Heater

Current State	Instruction	Next State	Action
[off,off]	SwitchOn	[on,off]	current!
[on,off]	TapOn	[on,on]	water!
⋮			

What is the point:

- While switching on, first turn the tap on and then the electricty.
- While switching off, first turn off the electricty and then the tap.

A more interesting example...

The state is a 2-tuple, such as [off,on], saying that electricity is off while water is on.

Water Heater

Current State	Instruction	Next State	Action
[off,off]	SwitchOn	[on,off]	current!
[on,off]	TapOn	[on,on]	water!
⋮			

What is the point:

- While switching on, first turn the tap on and then the electricty.
- While switching off, first turn off the electricty and then the tap.
- The state [on,off] is UNSTABLE.

Well, actually...

Example

Stapler:

Current State	Instruction	Next State	Action
Ready	Hit	Ready	staples the sheets
Ready	Hit	Jam	messed it up
Jam	Hit	Jam	more mess

Example

Vending Machine:

Current State	Instruction	Next State	Action
Ready	Tea	Ready	Vend Tea
Ready	Tea	Over	no more
Over	Tea	Over	why???
Ready	Coffee	Ready	Vend Coffee

Example

Tape Recorder: States: { FF, BB, play, idle, empty}
Instructions { FF, BB, Play, Stop, OpenDoor

Current State	Instruction	Next State	Action
FF	Stop	Idle	stops FF
Idle	Play	play	playing tape
play	Eject	error	block this

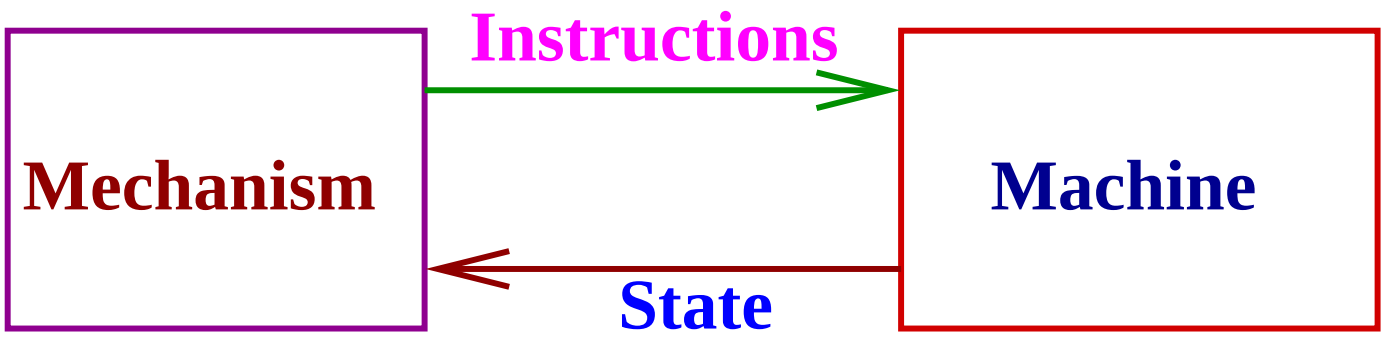
In summary:

- A machine has a set of states and instructions.
- At any moment, a machine is in one of the states.
- An instruction causes some action and a change of state.
- Not all instructions are executable for any state.

Mechanisms and Machines

Mechanism

A mechanism is a device of supplying instrcutions to a machine.



- Thus the mechanism *runs* the machine by issuing instructions.
- The machine executes this instructions and updates its state.
- The mechanism can access this state and issues the next instruction.

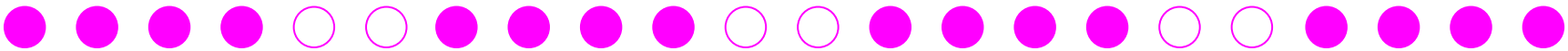
Actually, most modern machines internally are mechanisms. For example, the vending machine above, when you ask for Tea does the following:

now	Instruction	next	Remarks
1	Open Tea Powder Nozzle	2	allows powder into mixing container
2	Shut Tea Powder Nozzle	3	enough tea powder
3	Open Hot Water Nozzle	4	into mixing container
4	Shut Hot Water Nozzle	5	
5	Open Vending Nozzle	6	
6	Shut Vending Nozzle	1	

Of course, if the machine state returns Jammed at anytime, then the execution is suspended.

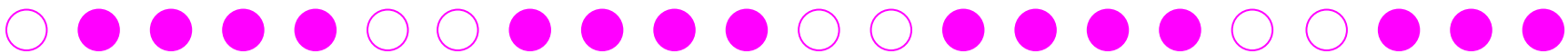
Fancy Lighting

Lets look at a more interesting mechanism.



Fancy Lighting

Lets look at a more interesting mechanism.



Fancy Lighting

Lets look at a more interesting mechanism.

Fancy Lighting

Lets look at a more interesting mechanism.

Fancy Lighting

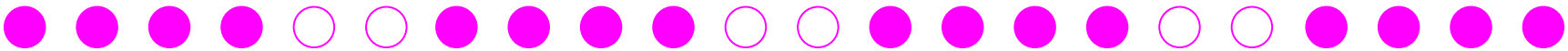
Lets look at a more interesting mechanism.

Fancy Lighting

Lets look at a more interesting mechanism.

Fancy Lighting

Lets look at a more interesting mechanism.



Fancy Lighting

Lets look at a more interesting mechanism.

Fancy Lighting

Lets look at a more interesting mechanism.

Fancy Lighting

Lets look at a more interesting mechanism.

Fancy Lighting

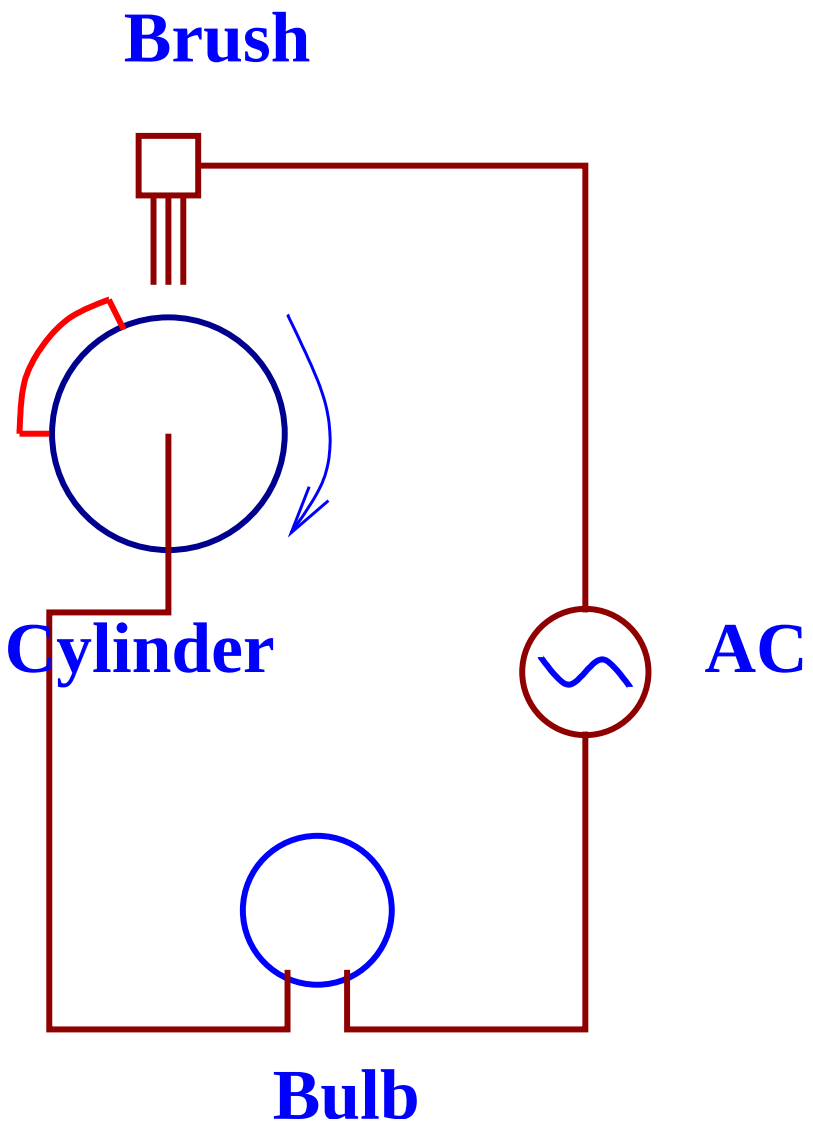
Lets look at a more interesting mechanism.

Fancy Lighting

Lets look at a more interesting mechanism.

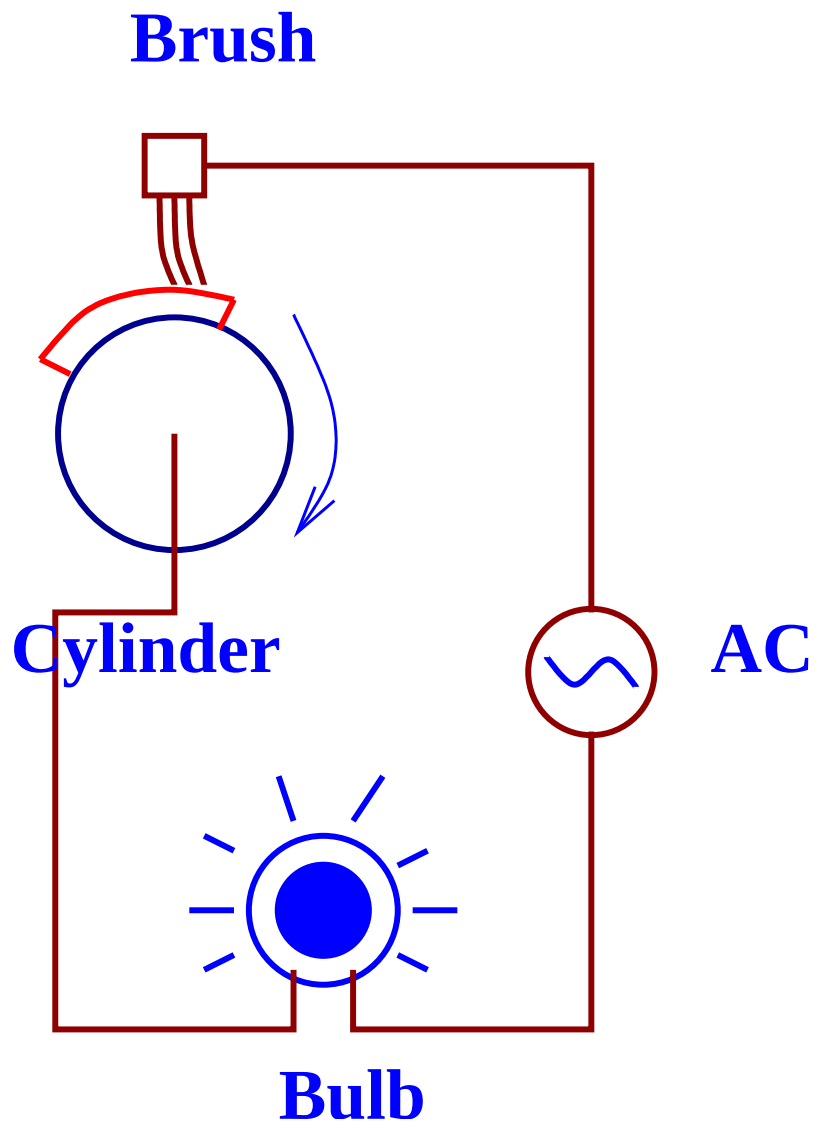
Fancy Lighting:How does it work?

- Bulbs are grouped together into (what we call) circuits.
- Each circuit has a brush.
- There is a rotating drum with portrusions.
- Portrusions on the drum establish electrical contact.
- The portruding parts on the circle decide when the lights of that circuit are on.



Fancy Lighting:How does it work?

- Bulbs are grouped together into (what we call) circuits.
- Each circuit has a brush.
- There is a rotating drum with portrusions.
- Portrusions on the drum establish electrical contact.
- The portruding parts on the circle decide when the lights of that circuit are on.



How do we describe this mechanism

1 22	create a 1-by-22 grid
1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4	the circuits
1 magenta	specifying colours
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	all the same colour

This describes the hardware of the system. There is an array of 1×22 bulbs with fixed bulb colours, and so on.

How do we describe this mechanism

1 1 1 1 0 0	light up ckts 1,2,3,4
0 1 1 1 1 0	
0 0 1 1 1 1	
1 0 0 1 1 1	
1 1 0 0 1 1	
1 1 1 0 0 1	the cycle ends
0.5	each time unit, in seconds

This is the program, which resides on the cylinder of the mechanism. As the cylinder rotates, the desired pattern emerges.

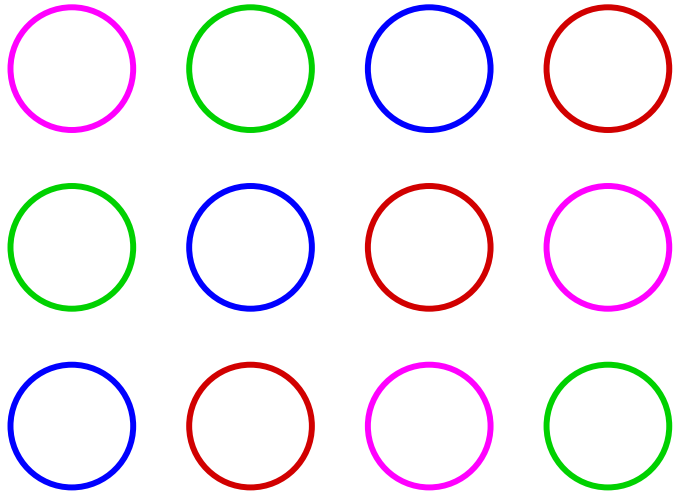
Program: A new concept

The sequence of instructions which are stored on the mechanism which drives the machine.

Another Example

The Hardware

3 4	create a 3-by-4 grid
1 1 1 1 2 2 2 2 3 3 3 3	the circuits
1 magenta 2 green 3 blue 4 red	specifying colours
1 2 3 4 2 3 4 1 3 4 1 2	the colour scheme



The Program

1 0 0 0 1 0 0 0 1	a 3-cycle
0.5	in seconds

Assignment 1

- Download the instructions, hardware and program files and execute the lighting program.
- Modify the above files to your taste and observe various combinations.
- Prepare a system which displays your roll-number, one digit after another.

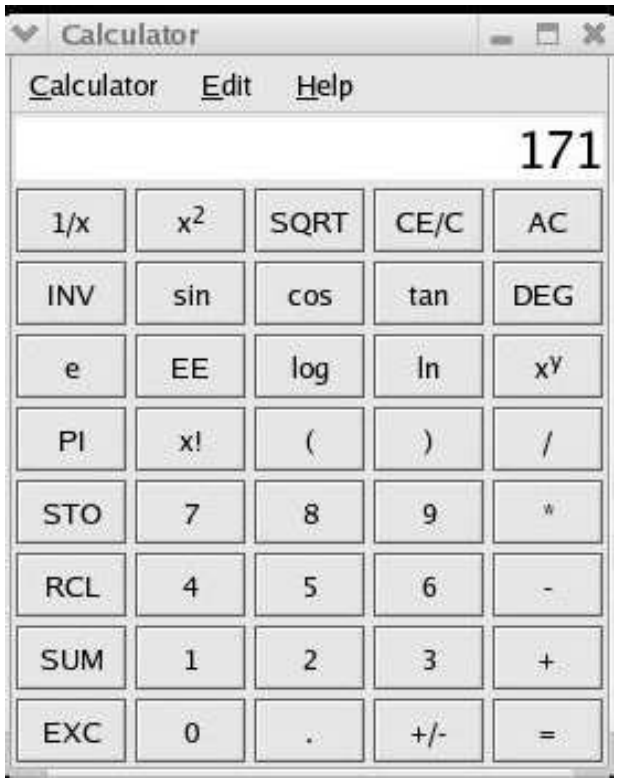
The Calculator as a Machine

The basic calculator

Types of keys:

- Unary operators such as cos, $\pm$,.
- Binary Operators such as $+, *$.
- Digits
- Memory related: `STO, RCL`
- `Special: = and AC.`

Then there is the display.



	action	display
1	3	3
2	+	3
3	2	2
4	=	5

Observation 1

The number 3 is stored internally and used during the binary operation. Let us call this as the internal register IR.

	action	display	IR
1	3	3	x
2	+	3	3
3	2	2	3
4	=	5	x

`x will stand for undefined.`

	action	display
1	3	3
2	+	3
3	2	2
4	=	5

Observation 1

The number 3 is stored internally and used during the binary operation. Let us call this as the internal register IR.

	action	display	IR
1	3	3	x
2	+	3	3
3	2	2	3
4	=	5	x

x will stand for undefined.

	action	display	IR
1	3	3	x
2	STO	3	x
3	2	2	x
4	+	2	2
5	RCL	3	2
6	=	5	x

Observation 2

The contents of the memory need to be stored as well. We call this as the M1 register.

action	display	IR	M1
3	3	x	x
STO	3	x	3
2	2	x	3
+	2	2	3
RCL	3	2	3
=	5	x	3

Lets look at another set of computations.

action	display	IR	M1
4	4	x	x
SQRT	2	x	x

and this:

action	display	IR	M1
3	3	x	x
+	3	3	x
4	4	3	x
SQRT	2	3	x
=	5	x	x

action	display	IR	M1
4	4	x	x
+	4	4	x
3	3	4	x
+	7	7	x
2	2	7	x
=	9	x	x

Observation 3

A unary operator is evaluated immediately. A pending binary operator is remembered.

Observation 3

A binary operator is evaluated as soon as its operands are specified.
Is this really true?

action	display	IR	M1
4	4	x	x
*	4	4	x
3	3	4	x
+	12	12	x
2	2	12	x
=	14	x	x

action	display	IR	M1
4	4	x	x
+	4	4	x
3	3	4	x
*	3	?	x
2	2	?	x
=	10	x	x

What is happening

The machine encounters a * which it decides must be evaluated before the +. So it has remembered 4 and +, and 3 when it encounters the * and 2.

However, we will not worry about the internals of a calculator right now. We all know how to use it. Let us learn how to make it even more useful.

Summary-The States of a Calculator

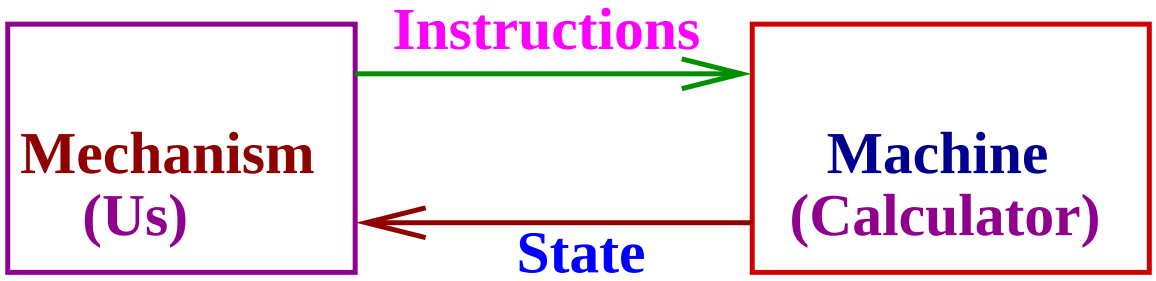
It is clear that the calculator remembers some operands and some operators. Let us assume that the calculator has the following four internal registers and 7 possible inputs:

D	The Display register
I	The Invisible register
M	The memory regsiter
O	The current operator

op1	a unary operator
op2	a binary operator
=	is equal to
num	a number
STO	memory store
RCL	memory recall
AC	All Cancel

The Calculator and the Program

Let us re-look at the machine-mechanism model and understand how we use a calculator.



We issue instructions and we observe the states.

instruction
23
+
10
=
STO
⋮

$\Longrightarrow$

D	*I*	*M*	*O*
23	x	x	x
23	23	x	+
10	23	x	+
33	x	x	x
33	x	33	x
⋮			

The Calculator and the Program

Program

A program is a sequence of instructions.

Problem: Write a program to convert centigrades to farenheit.

instruction	display
40	40
*	40
8	8
DIV	320
5	5
+	64
32	32
=	96

The Calculator and the Program

Program

A program is a sequence of instructions.

Problem: Write a program to convert centigrades to farenheit.

instruction	display
0	0
*	0
8	8
DIV	0
5	5
+	0
32	32
=	32

The Calculator and the Program

Program

A program is a sequence of instructions.

Problem: Write a program to convert centigrades to farenheit.

instruction	display
10	10
*	10
8	8
DIV	80
5	5
+	16
32	32
=	48

The Calculator and the Program

Program

A program is a saved sequence of instructions.

```
10    % substitute centigrades here
*
8
DIV
5
+
32
=     % observe the answer in the display
```

The above may be saved on a piece of paper or written on a computer file and re-used when necessary. It can be shared and transmitted. It can be written in Bangalore but executed in New York.

Another Program

Problem

Write a CAL-Program to compute the polynomial $p(t) = 3t^2 + 2t + 1$.

We use $3t^2 + 2t + 1 = (3 * t + 2) * t + 1$

```
1       % substitute t here
STO     % stored in memory
3
*
RCL
+
2
=       % 3t+2 is done
*
RCL
+
1
=       % read answer in display
```

Sample Executions

instruction	Display	Memory
2	2	x
STO	2	2
3	3	2
*	3	2
RCL	2	2
+	6	2
2	2	2
=	8	2
*	8	2
RCL	2	2
+	16	2
1	1	2
=	17	2

Indeed $p(2) = 17$.

Sample Executions

instruction	Display	Memory
0	0	x
STO	0	0
3	3	0
*	3	0
RCL	0	0
+	0	0
2	2	0
=	2	0
*	2	0
RCL	0	0
+	2	0
1	1	0
=	1	0

Indeed $p(0) = 1$.

In Summary

The Programmer

- Writes a CAL-program by assuming a typical input.
- Writes where typical inputs are to be replaced by user inputs.
- Stores/writes and transmits.

The Bum

- Receives the program.
- Substitutes his inputs.
- Runs the program on his calculator line-by-line in that order.

```
10   % substitute input here
*
8
DIV
5
+
32
=    % see output in display
```