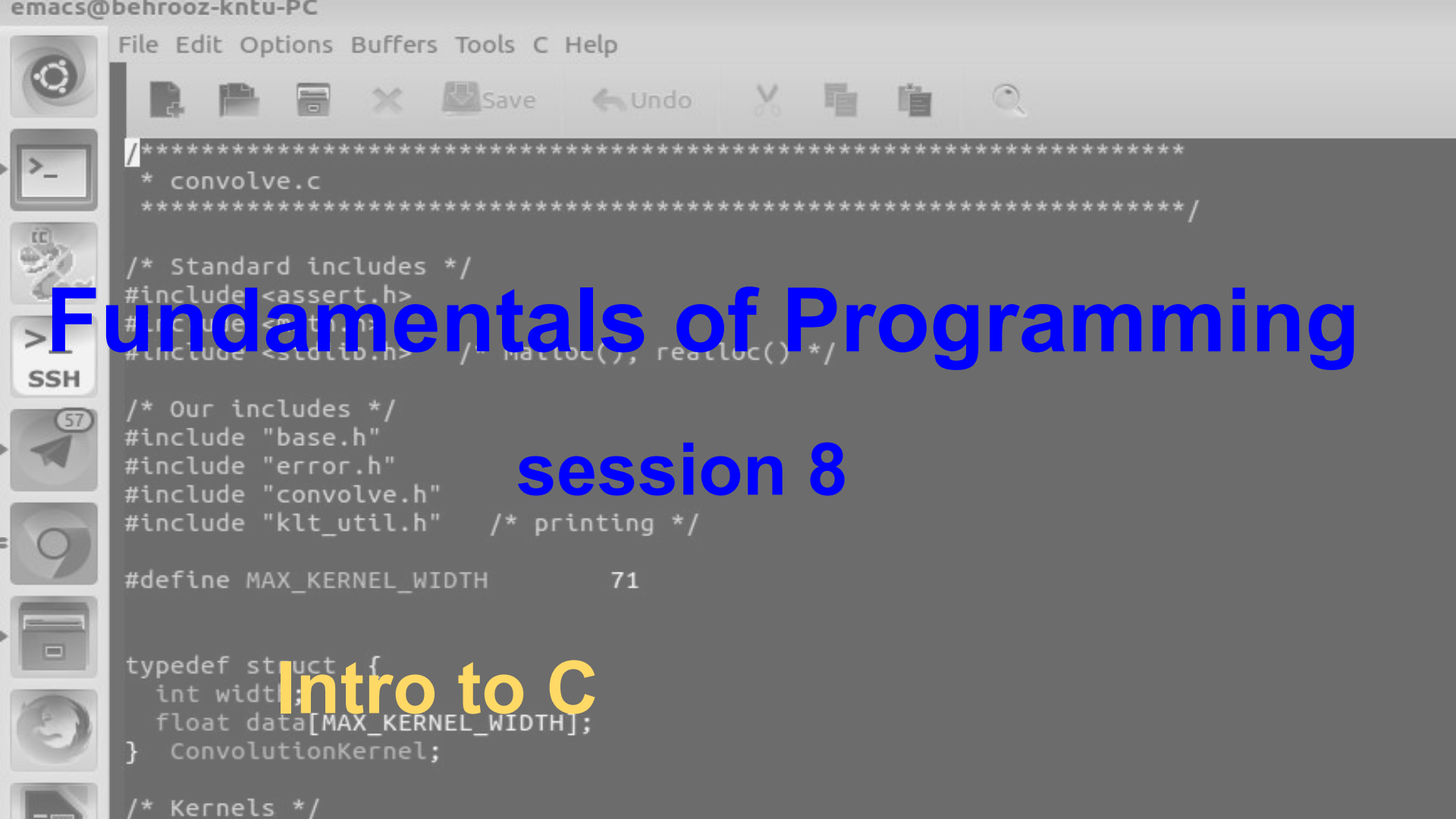
emacs@behrooz-kntu-PC
File Edit Options Buffers Tools C Help
Save
Undo
/***
* convolve.c
***/
/* Standard includes */
#include <assert.h>
#include <math.h>
#include <stdlib.h> /* malloc(), realloc() */
/* Our includes */
#include "base.h"
#include "error.h"
#include "convolve.h"
#include "klt_util.h" /* printing */
#define MAX_KERNEL_WIDTH 71
typedef struct {
int width;
float data[MAX_KERNEL_WIDTH];
} ConvolutionKernel;
/* Kernels */
Fundamentals of Programming
session 8
Intro to C

Binary Multiplication

Overflow

- Unsigned
- Signed

Your first (?) C program

```c
#include <stdio.h>

int main() {

  puts("Salam! Chetori!!!???");

  return 0;

}
```

A C program

```c
#include <stdio.h>

int main() {

  printf("Salam! Chetori!!!???");

  return 0;

}
```

A C program

```c
#include <stdio.h>

int main() {

  printf("Salam! Chetori!!!???\n");

  return 0;

}
```

A C program

```c
#include <stdio.h>

int main() {

  printf("Salam! Chetori!!!???\n\n\n\n");

  return 0;

}
```

Variables

```c
#include <stdio.h>

int main() {
  char a;
  signed char a1;
  unsigned char a2;

  short b;
  signed short b1;
  unsigned short b2;

  int c;
  unsigned int c1;
  signed int c2;


  long d;
  unsigned long d1;
  signed long d2;

  float f;
  double g;


  return 0;

}
```

Assignment (=) and printing variables

```c
#include <stdio.h>

int main() {
  int a;

  a = -1;

  printf("Salam %d Chetori?\n", a);

  return 0;

}
```

the **printf** function formats

%c	character
%d	decimal (integer) number (base 10)
%e	exponential floating-point number
%f	floating-point number
%i	integer (base 10)
%o	octal number (base 8)
%s	a string of characters
%u	unsigned decimal (integer) number
%x	number in hexadecimal (base 16)
%%	print a percent sign
\%	print a percent sign

https://alvinalexander.com/programming/printf-format-cheat-sheet

reading variables

```c
#include <stdio.h>

int main() {
  int a,b;

  scanf("%d",&a);
  scanf("%d",&b);


  printf("a=%d, b=%d, a+b=%d, a-b=%d\n", a,b, a+b, a-b);

  return 0;

}
```

decision making

```c
#include <stdio.h>

int main() {
  int a,b;

  scanf("%d",&a);
  scanf("%d",&b);

  if (a > b) {
    printf("%d\n", a);
  }

  return 0;

}
```

decision making

```c
#include <stdio.h>

int main() {
  int a,b;

  scanf("%d",&a);
  scanf("%d",&b);

  if (a > b) {
    printf("%d\n", a);
  } else {
    printf("%d\n", b);
  }

  return 0;

}
```

decision making

```c
#include <stdio.h>

int main() {
  int a,b;

  scanf("%d",&a);
  scanf("%d",&b);

  if (a > b) {
    puts("a is bigger than b");
  } else if (a < b) {
    puts("a is smaller than b");
  } else {
    puts("a equals b");
  }

  return 0;

}
```

Comparing variables

Operator	Description	Example
==	Checks if the values of two operands are equal or not. If yes, then the condition becomes true.	(A == B) is not true.
!=	Checks if the values of two operands are equal or not. If the values are not equal, then the condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand. If yes, then the condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand. If yes, then the condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand. If yes, then the condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand. If yes, then the condition becomes true.	(A <= B) is true.

Assignment (=) vs equals (==) operator