University of Washington Computer Programming I

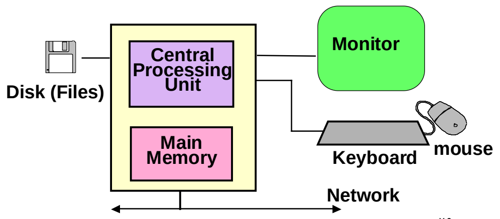
File Input/Output

Overview

Concepts
- External disk files
- Opening files for reading/writing
- File variables
- File I/O
- Closing Files

Reading
- Textbook sec. 12.1

Review: What is I/O?



Files Defined

A “file” is a collection of data on disk

Why Have Files?

Large volume of input data

Large volume of output data

More permanent storage of data

Transfer to other programs

Multiple simultaneous input and/or output streams

Data Files

Business Data: customer files, payroll files, …

Scientific Data: weather data, environmental data, topographic maps, …

Image Data: web images, satellite images, medical images, …

Web Data: HTML, GIF, JPEG, PNG, XML, …

Files vs. File Names

A “file” is a collection of data on disk
- Managed by the user and the operating system
- Permanent

A “file name” is how the user and OS know the file
- follows OS naming rules

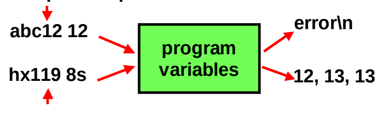
We'll look at using text files in a C program

V-7

Files as Streams of Characters

keyboard/screen are special cases

input / output streams of characters



Multiple streams can be used simultaneously

In reality, stream flows through a buffer rather than directly into or out of variables.

V-8

Files vs. File Variables

Key idea
- A file is a collection of data on disk
  - For our purposes, a sequence of characters
- But a C program can only operate directly on variables (data in main memory), so…
- Need to make a connection between the data on the disk and variables in main memory

V-9

Files vs. File Variables

A file variable is a data structure in the C program which represents the file
- Temporary: exists only when program runs

There is a struct called FILE in <stdio.h>
- Details of the struct are private to the standard C I/O library routines

File variables in C programs are pointers to a FILE *struct*.

V-10

*FILE *myfile;*

What’s in *stdio.h?*

Prototypes for I/O functions.

Definitions of useful *#define* constants
- Example: EOF for End of File

Definition of *FILE struct* to represent information about open files.

V-11

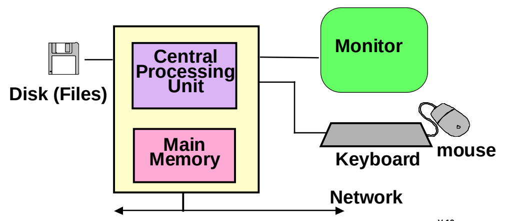
Opening A File

“Opening” a file: making a connection between the operating system (file name) and the C program (file variable)

Files must be opened before they can be used

V-12

Computer System



V-13

Opening A File

To open a disk file in C:
  library function *fopen*
  specify "r" (read, input) or "w" (write, output)
    NB String "r", not char 'r' !

Files stdin/stdout (used by scanf/printf) are automatically opened and connected to the keyboard and display

V-14

File Open Example

```
/*usually done only once in a program*/
/*usually done near beginning of program*/

FILE *infilep, *outfilep;  /*file variables*/
char ch;

/* Open input and output files */
infilep   = fopen ("Student_Data",         "r" ) ;
outfilep = fopen ("New_Student_Data", "w") ;
```

V-15

File I/O: *fscanf* and *fprintf*

Once a file has been opened...

use *fscanf* and *fprintf* to read or write data from/to the file

Use the file variable returned by *fopen* to identify the file to be read/written

File must already be open before *fscanf* or *fprintf* is used!

V-16

File I/O: *fscanf* and *fprintf*

fscanf: works just like *scanf*, but 1st parameter is a file variable
  fscanf (filepi, "%...", &var, ...) ;

fprintf: works just *printf*, but 1st parameter is a file variable
  fprintf (filepo, "%...", var, ...) ;

V-17

Copying a File

```
/* Copy a file one character at a time */
/* files must already be open before this*/

fscanf (infilep, "%c", &ch);
while ( /*...we just read a character .. */ ) {
    fprintf (outfilep, "%c", ch) ;
    fscanf (infilep, "%c", &ch);
    }
```

Question: How do we tell when we've read the last character from the input file?

V-18

Detour: *(f)scanf* return value

Both *scanf* and *fscanf* are really int-valued functions

Besides storing input values in variables, they return a status code:

- •Tells the number of values successfully read
- •Can be used to see if the number of values read is the number expected. If not, there must have been an error.
- •Can also be used to detect when end of file is reached

V-19

scanf Old Style

```
int status, id, score ;
double grade ;
scanf("%d %lf %d", &id, &grade, &score) ;
```

Problem: how would you know if there was an error in input (garbage typed when number expected, etc.)?

V-20

scanf with return value

```
int status, id, score ;
double grade ;
status = scanf("%d %lf %d", &id, &grade,
                                  &score) ;
if (status < 3)
    printf("Error in input \n") ;
```

This is how you can detect errors

Don't confuse status (function value) with input (variables in parameter list)

V-21

End of File (EOF) Status

EOF: a special status value
Returned by *scanf* and *fscanf* when end of data is reached

defined in *stdio.h*
#define EOF (some negative value)

I/O library routines use EOF in various ways to signal end of file.
    Your programs can check for EOF

EOF is a status, not an input value!!

V-22

File Copy Example, Concluded

```
/* Copy a file one char at a time until EOF*/
/* files must already be open before this*/
status =  fscanf (infilep, "%c", &ch);
while ( status != EOF ) {
    fprintf (outfilep, "%c", ch) ;
    status =  fscanf (infilep, "%c", &ch);
    }
```

V-23

Closing A File

Breaks the link between file variable and file
Usually done only once in a program, near end of program
Closing an output file is essential, or data may be lost!

```
    infilep   = fopen ("Student_Data", "r" ) ;
    .../*process the file */
    ...
    /*when completely done with the file:*/
    fclose (infilep);
```

V-24

File Copy (Compact Edition)

```
/* Many C programmers use this style*/
...
while ( fscanf (infilep, "%c", &ch) != EOF )
    fprintf (outfilep, "%c", ch) ;

printf ("File copied.\n") ;
fclose (infilep) ;
fclose (outfilep) ;
```

V-25

Review: Essential Functions for Text File I/O

fopen and fclose

fscanf:

```
status = fscanf (filepi, "%...", &var, ... ) ;
 /* fscanf returns EOF on end of file */
```

fprintf:

```
fprintf (filepo, "%...", var, ... ) ;
```

File must already be open before before fscanf or fprintf is used!

V-26

Building Applications with Files

With *fopen, fclose, fprintf*, and *fscanf* you can write lots of useful programs involving files

Many errors and exceptions can arise when using files

- A robust program must handle errors
- scanf's return value informs us of errors

V-27

Summary

Files are collections of data on disk

A file must be opened be used

A file should be closed after use

fprintf is used to write text files

fscanf is used to read text files

scanf/fscanf operation returns a status

V-28