

SWE4001 – System Programming
Module 7: Editors and Debugging system
Lesson 3 of 5: Editor Structure

TEXT EDITING OPERATIONS

- ❖ INSERT
- ❖ DELETE
- ❖ REPLACE
- ❖ MOVE
- ❖ COPY

OVERVIEW OF AN EDITING PROCESS

- ❖ Search and locate the area to be manipulated – TRAVELLING
- ❖ Select the required portion – FILTERING
- ❖ Specify the command for modifying the target document - EDITING PHASE
- ❖ Decide how to format the present view and how to display it - FORMATTING
- ❖ Edited text or document is displayed on the screen - VIEWING

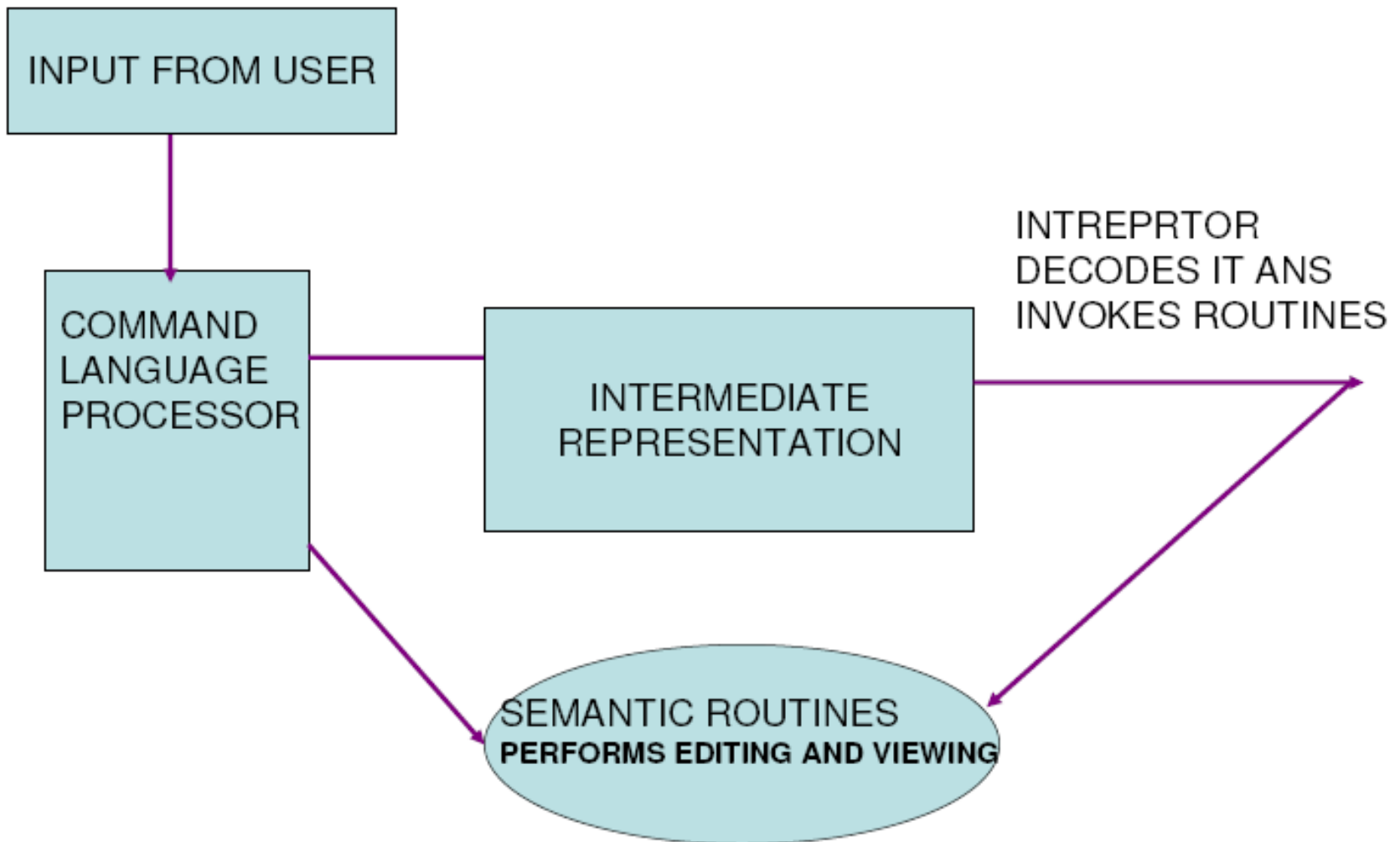
- ❖ These operations done with the help of traveling, editing and filtering component
- ❖ These components contain modules for that particular operation

Editor structure

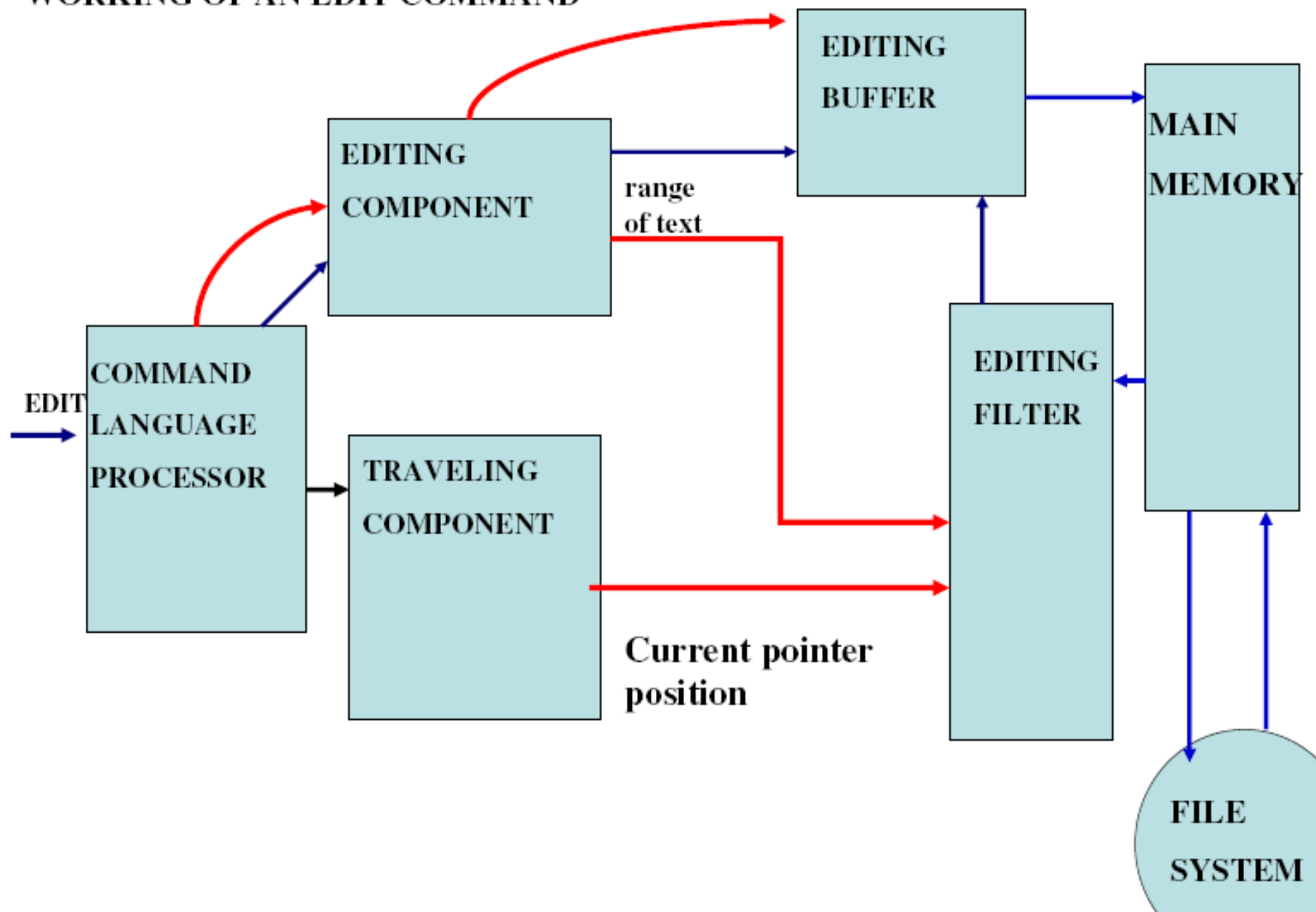
❖ Fundamental functions in editing are

- Traveling
 - Movement of the editing context to a new position within the text
- Editing
 - Formatting the text in a manner desired by the user
- Viewing
 - Formatting the text and prepare them for viewing
- Display
 - It gives the text in the user's requirements

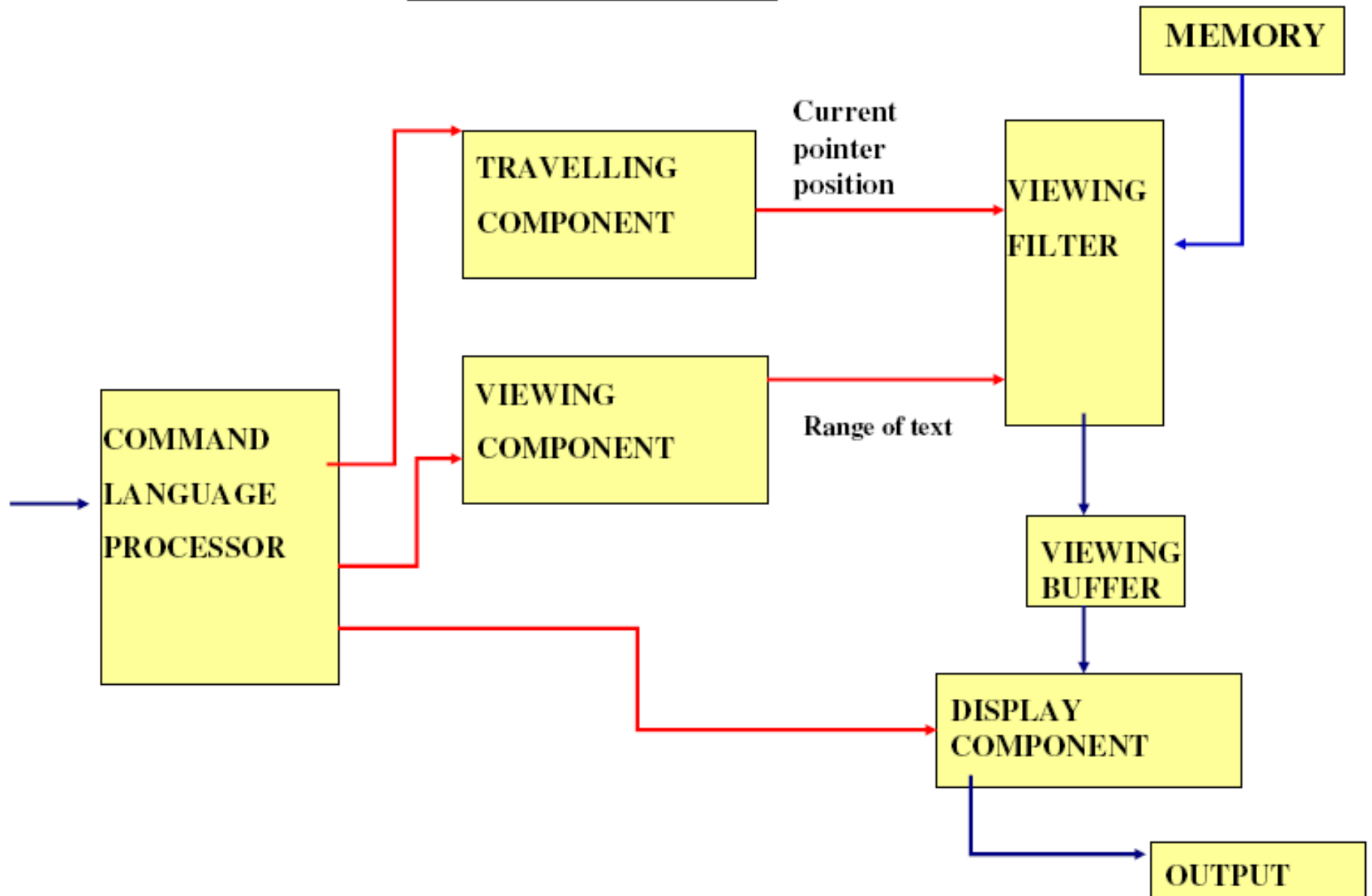
DESIGN OF AN EDITOR



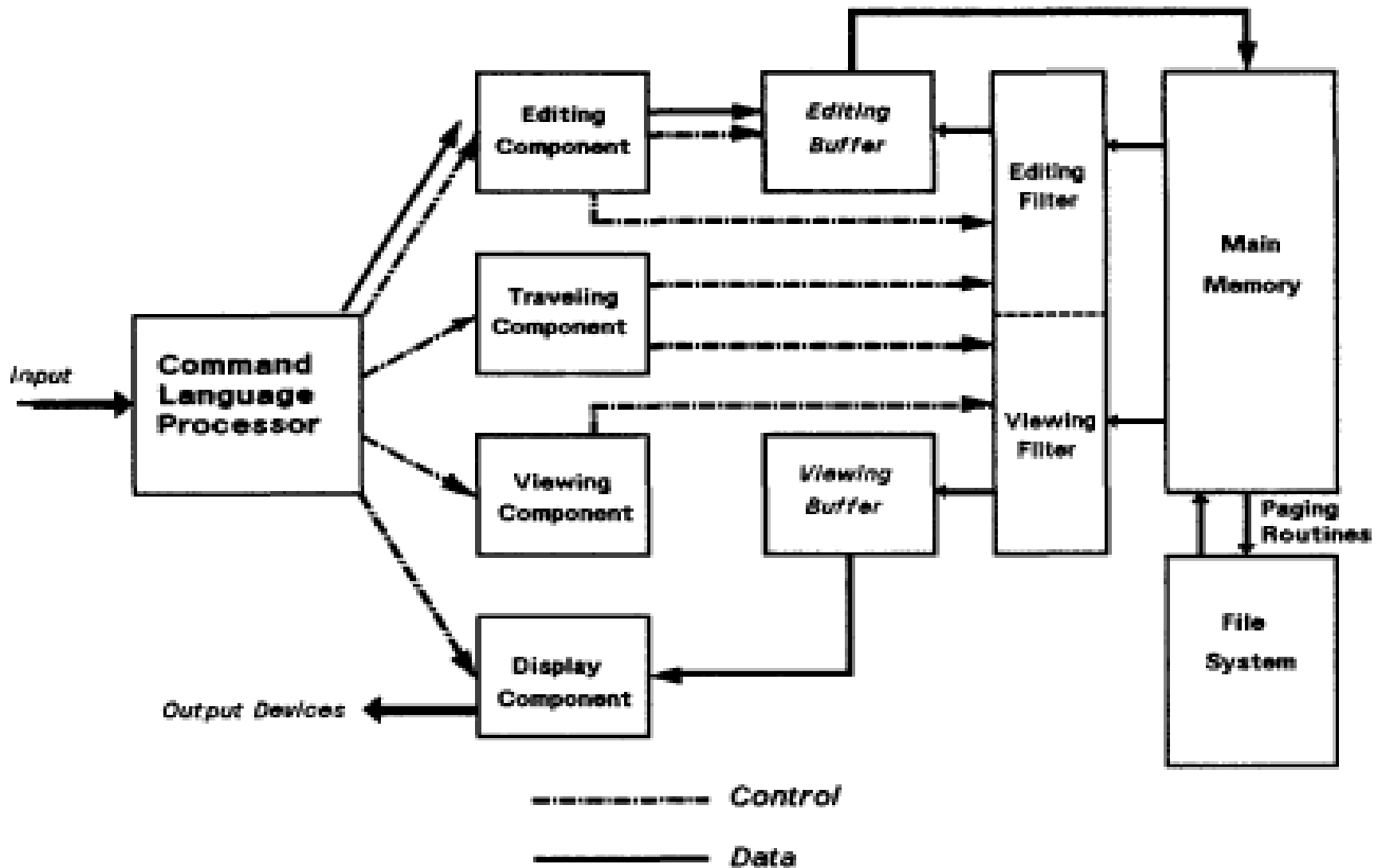
WORKING OF AN EDIT COMMAND



VIEWING PROCESS



Typical editor structure



Command language processor

- ❖ It accepts the input from user's input devices and analyses the tokens and syntactic structure of the commands.
- ❖ It produces an intermediate representation of the desired editing operations.

Command language processor

- ❖ Editing are always specified explicitly by the user and display operation are specified implicitly by the other three categories of operations
- ❖ Traveling and viewing operations may be invoked either explicitly by the user or implicitly by the editing operation.

Document editing

- ❖ The start of the area to be edited is determined by the **current editing pointer** maintained by the **editing component**
- ❖ **Traveling component** of the editor actually performs the setting of the current editing and viewing pointers

Document editing

- ❖ When the user issued the editing command, the **editing component** invokes the editing filter
- ❖ It filters the document to generate new editing buffer based on the current editing filter parameters

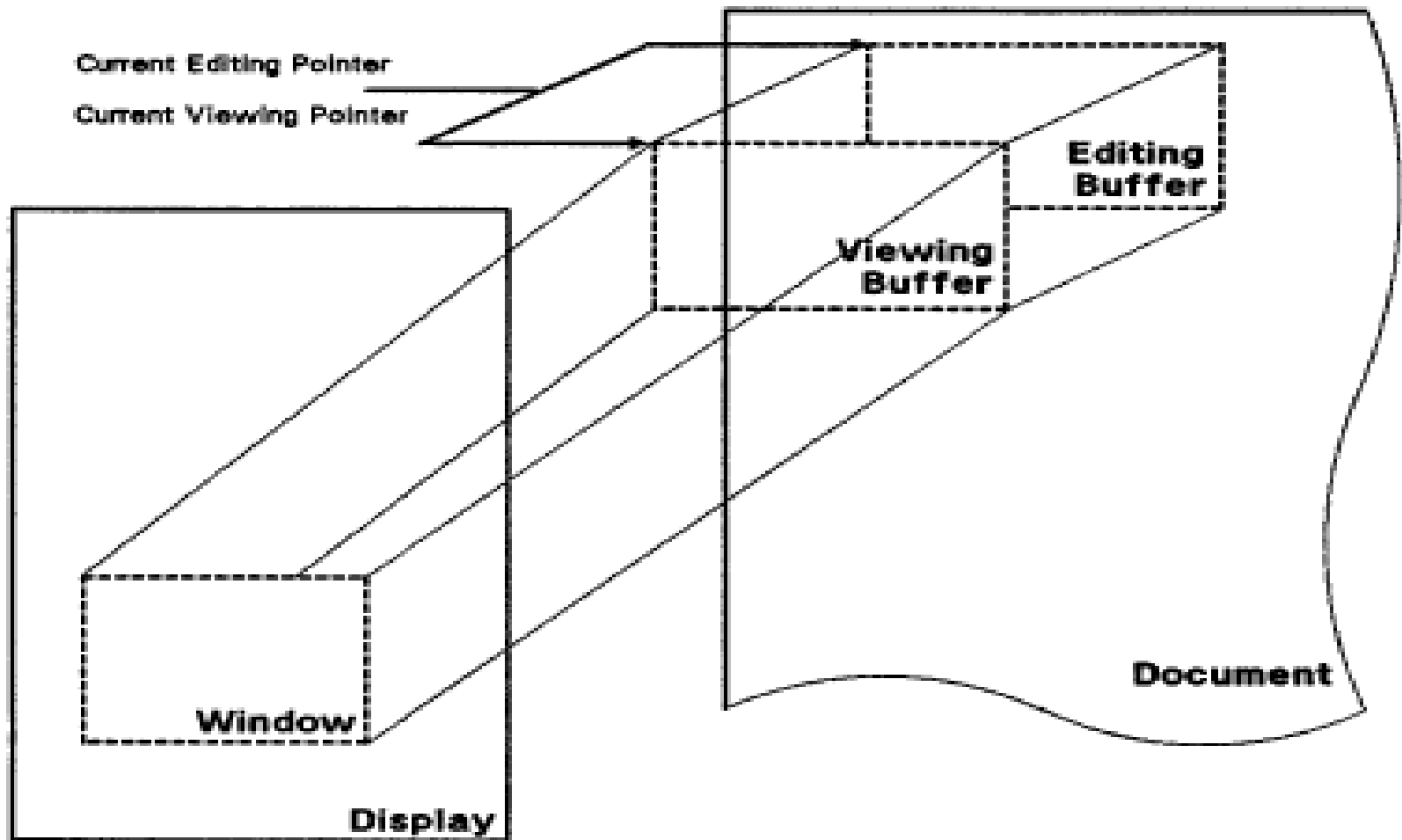
Document viewing

- ❖ The start of the area to be viewed is determined by the **current viewing pointer**.
- ❖ The current viewing pointer can be set or reset explicitly by the user with a traveling command or implicitly by the system as a result of the previous editing operation.
- ❖ When the display needs to be updated, the viewing component invokes the **viewing filter**

Document viewing

- ❖ This component filters the document to generate a new **viewing buffer** based on the current viewing pointer
- ❖ In line editors, the viewing buffer may contain the current line; in screen editors, this buffer may contain a rectangular cutout of the quarter-plane of text.
- ❖ This viewing buffer is then passed to the display component of the editor, which produces a display by mapping the buffer to a rectangular subset of the screen, usually called a window.

Simple relationship between editing and viewing buffers



Example

❖ [1,30] c/enit/edit/

Mapping of viewing buffer to window

❖ Two components of the system

- Viewing component formulates an ideal view, often expressed in a device independent intermediate representation.
- Display component takes the idealized view from the viewing component and maps it to a physical output device in the most efficient manner possible.

EDITORS IN 3 TYPES OF ENVIRONMENTS

❖ TIME SHARING

- editor should be very fast in obtaining functions from the host OS

❖ STAND ALONE

- editor should run independently on each machine

❖ DISTRIBUTED RESOURCE SHARING

- should have the functionality of TIME SHARING & STAND ALONE