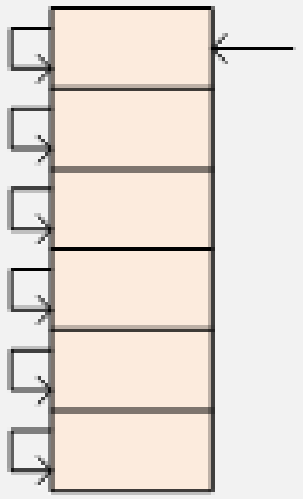
RANDOM AND DIRECT ACCESS

Two Terms, One Concept

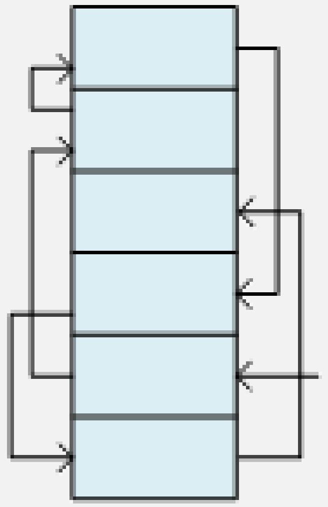
Delroy A. Brinkerhoff

RANDOM/DIRECT ACCESS

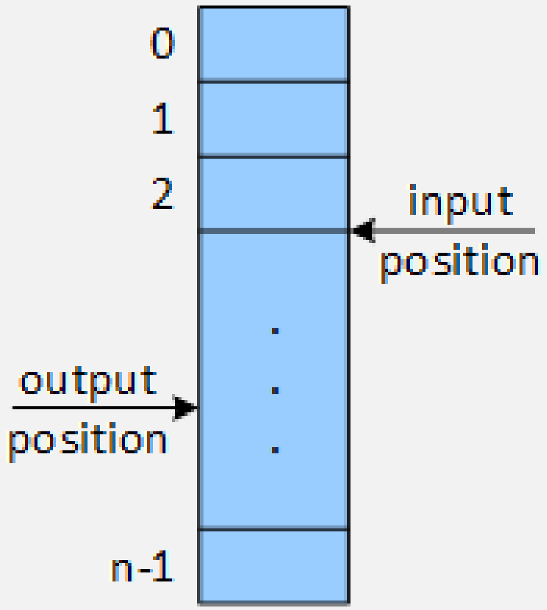
SEQUENTIAL ACCESS



RANDOM/DIRECT ACCESS



fstream OBJECTS HAVE TWO POSITION POINTERS

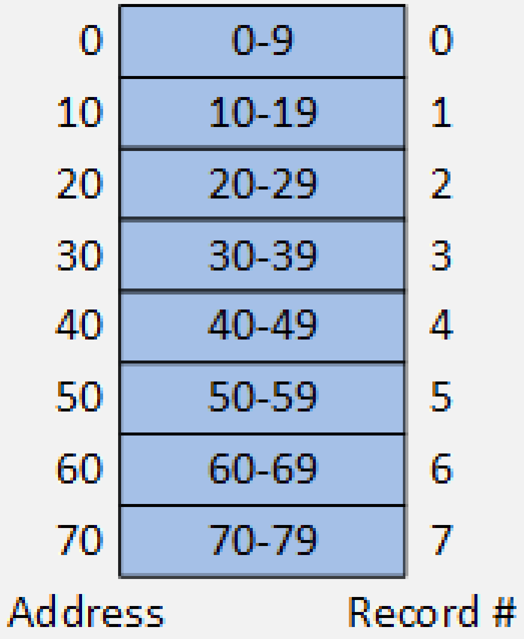


- `istream& seekg(streampos pos);`
- `ostream& seekp(streampos pos);`
- `istream& seekg(streampos off, seekdir loc);`
- `ostream& seekp(streampos off, seekdir loc);`
  - `ios::beg`
  - `ios::cur`
  - `ios::end`
- `streampos tellg();`
- `streampos tellp();`

THE RELATIONSHIP BETWEEN ADDRESSES AND RECORD NUMBERS

- Address (physical) ↔ Record number (problem)
- address = record number × size of a record
- record number = address / size of a record

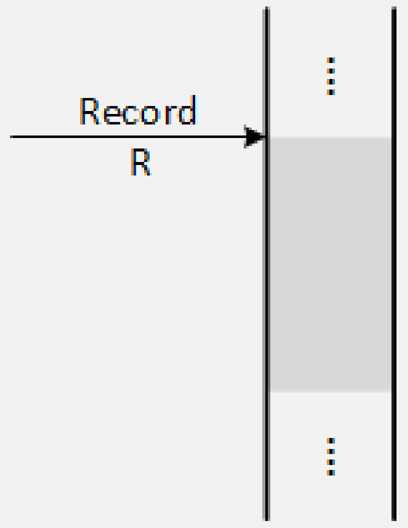
- `struct chunk { . . . };`
- `streampos offset = record * sizeof(chunk);`
- `streampos record = offset / sizeof(chunk);`



FILE POSITIONING OPERATIONS

Operation	Meaning
`s.seekg(0);`	Move read pointer to file's start
`s.seekp(0);`	Move write pointer to file's start
`s.seekg(0, ios::end);`	Move read pointer file's end
`s.seekp(0, ios::end);`	Move write pointer file's end
`s.seekg(R * sizeof(chunk));`	Move read pointer to record *R*
`s.seekp(R * sizeof(chunk));`	Move write pointer to record *R*

UPDATING A RECORD: THE FUNDAMENTAL DATABASE OPERATION



- `s.seekg(R * sizeof(chunk));`
- `s.read((char*)c, sizeof(chunk));`
- *update c*
- `s.seekp(R * sizeof(chunk));`
- `s.write((char*)c, sizeof(chunk));`