stdio

Henning Schulzrinne
Columbia University

stream model

- Low-level Unix (and Windows) I/O: numeric file descriptor (file handle)
- first for Unix, now ANSI C
- handles
  - buffer allocation: read into large buffer, dump to OS in fixed units
  - performs I/O in optimal-sized chunks
- usually, much more efficient than system calls (read, write)
  - fewer system calls

Streams

- stdio library manipulates streams
- "associate stream with file"
- works for files, but also interprocess communications
- fopen returns pointer to FILE object (*file pointer*)
  - file descriptor
  - pointer to buffer

Buffering

- minimal number of read() and write()
- fully buffered: I/O buffer is filled
- line-buffered: newline character
- unbuffered: as soon as possible

```
void setbuf(FILE *fp, char *buf); // BUFSIZE or NULL
void setvbuf(FILE *fp, char *buf, int mode, size_t
  size);
int fflush(FILE *fp);
```


Positioning a stream

- ftell() and fseek(): 32-bit offset

```
long ftell(FILE *fp);
int fseek(FILE *fp, long offset, int whence);
  // SEEK_SET, SEEK_CUR, SEEK_END
void rewind(FILE *fp);
```

- get and set (opaque position!)

```
int fgetpos(FILE *fp, fpos_t *pos);
int fsetpos(FILE *fp, const fpos_t *pos);
```