

# オペレーティングシステム講義資料 6

## ファイルシステム API とメモリマップドファイル

田浦

## 1 ファイルシステム API

### 1.1 基本概念

- 開く
- 読む・書く
- 位置あわせ(seek)
- 閉じる

### 1.2 実例

- Unix: open, read/write, lseek, close
- Windows : CreateFile, ReadFile/WriteFile, SetFilePointer, CloseHandle
- 標準 C ライブラリ : fopen, fread/fwrite/fgetc/fputc/fgets/fputs/fscanf/fprintf, fseek, fclose

## 2 メモリマップドファイル

### 2.1 基本概念

- ファイルをメモリの様に読み書きする API
- Unix : open + mmap
- Windows : CreateFile + CreateFileMapping + MapViewOfFile

### 2.2 共有マップとプライベートマップ

- 共有マップ: 同一ファイルの同一の場所への複数の共有マップは、書き込み結果を共有(物理メモリを共有). また、ファイルへ書き込みを反映する
- 共有マップは「プロセス間共有メモリ」としても利用可能
- プライベートマップ: 同一ファイルの同一場所へのマッピングであっても、書き込み結果は共有されない(物理メモリはマッピングごとに別々に割り当てられる). ファイルへも反映されない

### 2.3 実現方式

- ファイルがメモリにマップされた時点では中身はメモリにコピーされているわけではない

- 初めてのアクセス時にページフォルトがおき、その際に OS がファイルの適切な場所から内容をメモリにコピーする(demand paging 機構の小さな一般化)

## 2.4 最適化

- 同じ場所に対する複数の読み出し専用(書き込み不可)マッピングは、(当然)、プライベートマッピングであっても物理メモリを共有してよい
- プログラムテキスト(共有ライブラリ)読み出し時に用いられている
- 共有マッピングであっても、実際に書き込みが起きるまでは物理メモリを共有してよい(copy-on-write)