[fopen 2](#_Toc399873288)

[fseek 3](#_Toc399873289)

[fread 4](#_Toc399873290)

[fwrite 5](#_Toc399873291)

[feof 6](#_Toc399873292)

[Ftell 7](#_Toc399873293)

[fclose 8](#_Toc399873294)

# fopen

## Voorbeeld

FILE\* tFile = NULL;

char\* tFileName = "bestand.dat";

tFile = fopen(tFileName, "rb");

## Resultaat

FILE = Indien het bestand goed geopend is, bevat deze variable de file pointer. Indien het openen mislukt wordt de variable NULL.

Errno = Indien het openen van het bestand mislukt staat op deze variable (globale variable) de foutcode. Deze kun je printen met “strerror(errno)” uit libraries “<errno.h>” en “<string.h>”.

## Toelichting

rb = Read Binary (bestaand bestand)

rb+ = Read & Write Binary (bijwerken vanaf begin vanaf bestand, bestand moet bestaan)

wb = Write Binary (nieuw bestand)

wb+ = Read & Write Binary (bestand aanmaken of overschrijven)

ab =Write Binary (bestand openen of aanmaken)

ab+ = Read & Write Binary (bestand openen en toevoegen aan eind van bestand of nieuw aanmaken)

­­­

# fseek

## Voorbeeld

FILE\* tFile = NULL;

char\* tFileName = "bestand.dat";

tFile = fopen(tFileName, "rb");

int tResult = -1;

int tSize = sizeof(int);

int tPosition = 10 \* tSize;

tResult = fseek(tFile, tPosition, SEEK\_SET);

## Resultaat

Indien het setten van de filepointer gelukt is wordt tResult 0 anders iets anders dan 0;

## Toelichting

SEEK\_SET = Positie vanaf het begin van het bestand

SEEK\_CUR = Positie vanaf de huidige positie, met – positie ga je naar voor en met + ga je naar achter

SEEK\_END = Positie vanaf het eind van het bestand

Een alternatief voor fseek(tFile,0,SEEK\_SET);

Is rewind(tFile); dan zet je de file pointer terug naar het begin van het bestand.

# fread

## Voorbeeld

FILE\* tFile = NULL;

char\* tFileName = "bestand.dat";

tFile = fopen(tFileName, "rb");

int tResult = -1;

STUDENT tStudent = { 0, “”, 0};

int tSize = sizeof(STUDENT);

int tPosition = 10 \* tSize;

fseek(tFile, tPosition, SEEK\_SET);

tResult = fread(&tStudent,tSize,1,tFile);

## Resultaat

tResult levert het aantal ingelezen elementen terug. Anders retourneert tResult 0

## Toelichting

Fread(ARG1,ARG2,ARG3,ARG4)

ARG1 🡪 pointer naar de variable waar je de gegevens wilt opslaan.

ARG2 🡪 grootte van 1 element.

ARG3 🡪 aantal elementen die je wilt lezen.

ARG4 🡪 file pointer naar het bestand waaruit je wilt lezen.

# fwrite

## Voorbeeld

FILE\* tFile = NULL;

char\* tFileName = "bestand.dat";

tFile = fopen(tFileName, "wb");

int tResult = -1;

STUDENT tStudent = { 0, “”, 0};

int tSize = sizeof(STUDENT);

int tPosition = 10 \* tSize;

fseek(tFile, tPosition, SEEK\_SET);

tResult = fwrite(&tStudent,tSize,1,tFile);

## Resultaat

tResult levert het aantal geschreven elementen terug. Anders retourneert tResult 0

## Toelichting

fwrite(ARG1,ARG2,ARG3,ARG4)

ARG1 🡪 pointer naar de variable waarvan je de gegevens wilt opslaan.

ARG2 🡪 grootte van 1 element.

ARG3 🡪 aantal elementen die je wilt schrijven.

ARG4 🡪 file pointer naar het bestand waarin je wilt schrijven.

# feof

## Voorbeeld1

FILE\* tFile = NULL;

char\* tFileName = "bestand.dat";

tFile = fopen(tFileName, "wb");

int tResult = -1;

tResult = feof(tFile);

## voorbeeld2:

while (!feof(tFile))

{

;

}

## Resultaat

tResult levert een 0 waarde als het EOF teken niet gevonden is.

## Toelichting

feof(ARG1)

ARG1 🡪 file pointer naar het bestand wat je wilt controleren.

# Ftell

## Voorbeeld1

FILE\* tFile = NULL;

char\* tFileName = "bestand.dat";

tFile = fopen(tFileName, "wb");

int tResult = -1;

tResult = ftell(tFile);

## Resultaat

tResult levert de huidige van de file pointer binnen het bestand ( vanaf het begin) Indien deze functie mislukt dan tResult -1;

## Toelichting

feof(ARG1)

ARG1 🡪 file pointer naar het bestand wat je wilt controleren.

# fclose

## Voorbeeld1

FILE\* tFile = NULL;

char\* tFileName = "bestand.dat";

tFile = fopen(tFileName, "wb");

int tResult = -1;

tResult = fclose(tFile);

## Resultaat

tResult levert 0 indien gelukt anders iets anders dan 0. (afhankelijk van de positie van de EOF indicator).

## Toelichting

feof(ARG1)

ARG1 🡪 file pointer naar het bestand wat je wilt controleren.