

Fil information

För att läsa en fil i C behöver du oftast ta reda på filens storlek innan du börjar läsa så du vet storleken på din buffert. Detta kan göras på många sätt och om du vill använda rå C kod kan man använda sättet jag valde vilket använder sig av **fseek**, **ftell** och **rewind** för att få tag på storleken (Läs bilaga 1). Detta är dock begränsat och ger bara storleken. För samma sak men OS-baserade funktioner kan du även i Windows använda till exempel **GetFileSize** eller **GetFileSizeEx**. Dessa är dock begränsade till platformen Windows och vill du då ta reda på samma sak fast på ett Unix baserat system kommer det inte att funka, då använder du istället funktionen **stat**.

Exempel Windows allokerar minne för fil till sträng:

```
HANDLE hFile = CreateFile("Path/To/File",
    GENERIC_READ,
    FILE_SHARE_READ,
    NULL,
    OPEN_EXISTING,
    FILE_ATTRIBUTE_NORMAL,
    NULL);

LARGE_INTEGER size;

if(hFile != INVALID_HANDLE_VALUE)
{
    BOOL bSuccess = GetFileSizeEx(hFile, &size);
}

char* file = malloc(sizeof(char) * size + 1);
file[size + 1] = '\0';
```

Exempel Unix allokerar minne för fil till sträng:

```
#include <sys/stat.h>

struct stat st;
stat("Path/To/File", &st);

char* file = malloc(sizeof(char) * st.st_size + 1);
file[st.st_size + 1] = '\0';
```

Utöver dessa finns det ju såklart saker man måste göra för att säkerställa att man inte får overflow och liknande dock så ser vi ju att det är rätt krångligt alltså föredrar jag sättet jag beskriver i bilaga 1.