**Beskrivning av den logiska strukturen i filsystemet**

Vårt filsystem bygger främst i grunden på vår abstrakta klass FS\_item. Alla filer och mappar i systemet ärver från denna klass och en folder kan hålla ett godtyckligt antal FS\_items. En folder ansvarar för skapande och borttagande av allt innehåll i just den foldern, samt eliminering av detta innehåll vid radering eller formatering. När ett filsystem byggs upp skapas en origin-mapp med namnet ”root”. Denna mapp kan inte raderas manuellt och är roten till trädet som bygger upp filsystemet.

En fil innehåll endast ett namn samt ett blocknummer, som korresponderar till ett blocknummer i memblockdevice som innehåller filens data.

**Beskrivning av alla funktioner i Filesystem**

FS\_item\* validFilePath(std::vector<std::string> filePath);

Detta är en privat medlemsfunktion som anropas varje gång en annan funktion är i behov av en filepath. Den börjar med att söka i root-mappen efter den första parametern i filePath, varefter den fortsätter djupare in mappar i sökvägen om den kan hitta dem tills den hittar det önskade objektet, antingen en fil eller en mapp, och returnerar en pekare till detta objekt. Om en tom filePath anges returneras en pekare till root-mappen. Om funktionen inte hittar det angivna sökobjektet returneras nullptr.

Std::vector<std::string> parseFilePath(std::string filePath)

Detta är en privat medlemsfunktion som tar emot en filepath i form av en std::string och omvandlar den till en std::vector<std::string> utan / eller \ som validFilePath()-funktionen kan arbeta med.

Void format()

Denna funktion anropas när användaren skriver format i shell. Funktionen anropar destruktorn till rootmappen som i sin tur kommer att anropa destruktorer till alla underliggande mappar och filer. Funktionen ersätter även mMemBlockDevice med en tom MemBlockDevice.

Bool createFile(std::string filePath, std::string fileContent)

Denna funktion anropas när användaren skriver ”create <filePath>” följt av önskat filinnehåll. Funktionen anropar validFilePath för att verifiera att mappen filen ska skapas i existerar, och sedan anropar den mappen för att se så att en fil eller mapp inte redan existerar med samma namn. Om den uppfyller alla krav letas ett tomt block upp i mMemBlockDevice och filinnehållet skrivs till det blocket.

Bool createFolder(std::string filePath)

Denna funktion anropas när användaren skriver “mkdir <filePath>”. Resterande funktionsbeteende är mycket likt det som beskrivs i createFile.

Bool removeFile(std::string filePath)

Denna funktion anropas då användaren skriver “rm <filePath>”. Funktionen anropar validFilePath för att verifiera att filen finns och anropar sedan removeFile i mappen som filen ligger i samt tömmer det block som filen ockuperade.

Bool goToFolder(std::string &filePath)

Denna funktion anropas då användaren skriver “cd <filePath>”. Funktionen anropar validFilePath för att verifiera att mappen finns, och sedan sätter den variabeln currentDirectory till den nya destinationsmappen. Detta används sedan i andra funktioner för att skapa filer lokalt i den nuvarande mappen.

Bool copyFile(std:.string filePath, std::string filePath2)

Denna funktion anropas då användaren skriver “cp <filePath1> <filePath2>”. Funktionen skapar en kopia av filen som specificeras i filePath1 till den sökväg som anges i filePath2, där även namnet på den nya filen är inkluderat. Genomförande är mycket likt det som beskrivs till funktionen createFile().

Std::string listDir()

Denna funktion anropas då användaren skriver ”ls”. Den returnerar namnen på alla filer och mappar som ligger i den mapp som är nuvarande ”current directory”.

Bool createImage(std::string filePath)