

# Protokoll für myfind

„main“: Das Programm beginnt in der „main“-Funktion. Es verarbeitet die Befehlszeilenargumente mit „getopt“ und setzt die Optionen „recursive“ und „caseInsensitive“ entsprechend. Es überprüft dann, ob der angegebene Suchpfad existiert, indem es die Funktion „exists()“ aus der „<filesystem>“-Bibliothek verwendet. Je nach Anzahl der zu suchenden Dateien wird entweder die Funktion „findFile“ oder „findFiles“ aufgerufen.

„findFiles“: Diese Funktion nimmt einen Suchpfad, eine Liste von Dateinamen, einen Booleschen Wert für die Rekursion und die Groß-/Kleinschreibung als Eingabe. Sie erzeugt für jede zu suchende Datei einen eigenen Kindprozess mit „fork()“. In jedem Kindprozess wird die Funktion „findFile“ aufgerufen.

„findFile“: Diese Funktion nimmt den Suchpfad, den Dateinamen, einen Booleschen Wert für die Rekursion und die Groß-/Kleinschreibung und optional eine Pipe als Eingabe. Sie durchläuft alle Dateien im angegebenen Verzeichnis (rekursiv oder nicht, abhängig von der gewählten Option) und ruft für jede Datei die Funktion „checkFile“ auf.

„checkFile“: Diese Funktion nimmt einen Verzeichniseintrag, einen Booleschen Wert für die Groß-/Kleinschreibung, einen Dateinamen und optional eine Pipe als Eingabe. Sie überprüft, ob der Dateiname der gesuchten Datei entspricht (unter Berücksichtigung der Groß-/Kleinschreibung). Wenn eine Übereinstimmung gefunden wird, wird entweder der absolute Pfad zur Datei direkt ausgegeben oder weiter verarbeitet.

Jeder Kindprozess durchläuft das Verzeichnis ähnlich wie in der Funktion „findFile“ beschrieben. Die Ergebnisse werden jedoch über eine Pipe an den Elternprozess zurückgegeben. Der Elternprozess liest die Ergebnisse aus der Pipe mit „read()“ und gibt sie aus. Nachdem alle Kindprozesse ihre Suche abgeschlossen haben, wartet der Elternprozess auf ihre Beendigung mit „waitpid()“.