

# Protokoll zur Programmieraufgabe

## myfind

von Stadler Alexander und Krzanovic Aleksandar

### ***Parallelisierung***

Als Konzept zur Parallelisierung verwenden wir in unserem Programm den UNIX-Befehl `fork()`. Damit wird jedes Mal, wenn das Programm nach einem Dateinamen sucht, ein neuer Kindprozess erstellt. Jeder Kindprozess führt dann, unabhängig vom Elternprozess, die Suche nach der Datei im angegebenen Verzeichnis durch. Auf diese Weise wird die Dateisuche effizient beschleunigt, da unter Verwendung von `fork()` die Suchprozesse parallel ausgeführt werden. Mit Hilfe des UNIX-Befehls `wait()` wartet der Elternprozess auf die Beendigung der Kindprozesse, sodass keine Zombie-Prozesse als Programmrest übrigbleiben.

### ***Synchronisation***

Wenn mehrere Kindprozesse gleichzeitig ausgeführt werden, so besteht das Risiko, dass die Ausgaben der Ergebnisse in der Standardausgabe (`stdout`) durcheinander geraten. Hierfür verwenden wir den Befehl `flock()` zur Dateisperre, um den Zugriff auf die Standardausgabe zu synchronisieren. Der Kindprozess sperrt die Standardausgabe, bevor er das Ergebnis ausgibt, und gibt die Sperre wieder frei, nachdem die Ausgabe abgeschlossen ist. Auf diese Weise verhindern wir asynchrone Ausgaben der Kindprozesse.

### ***Rekursive Suche***

Mit der Option `-R` wird die Dateisuche durch die `search_file()` Methode rekursiv auf Unterverzeichnisse ausgeweitet.

### ***Case-Sensitivity***

Mit der Option `-i` kann man unabhängig von Groß-/Kleinschreibung nach Dateien suchen. Hierfür verwenden wir die `equal()` Funktion aus der Standardbibliothek.