

Chris Bielow

Institut für Informatik
Bioinformatics Solution Center

Praktikum: Algorithmen und Datenstrukturen

Erste Programmieraufgabe

| |
|--|
| Abgabe Montag, 6.11., 23:59 Uhr via GIT |
|--|

Vorarbeit Tragen Sie sich bis Do, 26.10.2023 23:55 Uhr im KVV -> Bereiche in ein Tutorium und eine Gruppe (Lab<X>) ein (siehe VL-Skript). Sie erhalten dann in den darauf folgenden Tagen Zugang zum Git-Repository.

Programmieraufgabe In dieser Programmieraufgabe werden Sie die exakte Stringsuche für Strings aus dem ASCII-Alphabet mithilfe des HORSPOOL Algorithmus implementieren.

Laden Sie sich das Material zu Aufgabe 1 aus dem KVV/Whiteboard herunter.

aufgabe1.zip enthält:

- aufgabe1.hpp Eine Header-Datei mit dem Interface, welches von ihnen implementiert werden soll. Jede Funktion ist kurz beschrieben und ihre Implementierung sollte genau diese Funktionalität abdecken. Einige Funktionen hängen semantisch von anderen ab. Erweitern Sie die aufgabe1.h um evtl. benötigte Klassenmember (Daten und Funktionen) und #include s. In der Datei sind die entsprechenden Stellen kommentiert.

- aufgabe1_main.cpp Eine Source-Datei mit main-Funktion. Grundlage für das ausführbare Programm, welches Sie mit Testdaten füttern koennen. Dieses Programm bekommt zwei Parameter beim Aufruf via Kommandozeile:

Den Text, und das gesuchte Pattern. z.B. :

```
> ./aufgabe1_main "This is a long text, is it?" "is"
```

- aufgabe1_test.cpp Eine Source-Datei mit main-Funktion, welche ihre Implementierung testet (wird zum Tutorium online gestellt).

Entpacken Sie alle Dateien in ein Unterverzeichnis 'aufgabe1' Ihres Git-Gruppen-Verzeichnisses. Falls sie noch keinen Git clone gemacht haben, benutzen Sie irgendein lokales Verzeichnis und kopieren Ihre Arbeit später in das Git-Verzeichnis hinein.

In den Source-Dateien ist der Programmfluss bereits vorgegeben. Ihre Aufgabe besteht darin, die vorgegebenen Funktions-Interfaces aus aufgabe1.hpp in einer 'aufgabe1.cpp' zu implementieren (diese Datei existiert noch nicht; legen Sie sie an; achten Sie darauf, diese Datei GENAU SO zu benennen; keine extra Zeichen, Gross/Kleinschreibung beachten!!!!).

Sie können ihre Programme folgendermaßen auf den Poolrechnern/Computeservern kompilieren und ausführen (vorher noch das Git Repository einmalig auf dem Zielrechner clonen; wenn Sie das Git-Repo in ihr \$HOME Verzeichnis clonen, haben sie von allen Rechnern des Fachbereichs darauf Zugriff):

aufgabe1_main.cpp:

```
> cd /path/to/local/gitrepo/groupX/aufgabe1
> g++ -std=c++17 -Wall -pedantic -O3 aufgabe1.cpp aufgabe1_main.cpp -o aufgabe1_main
> ./aufgabe1_main "Yet another text?" "really?"
> echo $?
```

aufgabe1_test.cpp:

```
> cd /path/to/local/gitrepo/groupX/aufgabe1
> g++ -std=c++17 -Wall -pedantic -O3 aufgabe1.cpp aufgabe1_test.cpp -o aufgabe1_test
> ./aufgabe1_test
> echo $?
```

Diese beiden Programme müssen unter Linux auf den Poolrechnern/Computeservern compilieren (wie oben beschrieben) und erfolgreich ausgeführt werden können (das 'echo \$?' sollte jeweils 0 zurückgeben).

Abgabe Legen Sie die gesuchten Dateien `aufgabe1.cpp` und `aufgabe1.h` im Unterverzeichnis `./aufgabe1/` ihres Gruppenverzeichnis an und checken Sie es ins GIT ein. Achten Sie auf korrekte Gross/Kleinschreibung (d.h. alles klein!).

Alle anderen Dateien aus der `aufgabe1.zip` sollten Sie ebenfalls ins Git einchecken, damit sie alle nötigen Dateien für eine Kompilation jederzeit bereit haben. Für die Bepunktung ist das jedoch nicht von Bedeutung.

Testen Sie, ob die URL (hier EXAMPLARISCH für Lab4) die notwendigen Dateien anzeigt!

```
https://git.imp.fu-berlin.de/adp2023/group04/blob/master/aufgabe1/aufgabe1.cpp
https://git.imp.fu-berlin.de/adp2023/group04/blob/master/aufgabe1/aufgabe1.h
```

Punkte Für diese Aufgabe es insgesamt 10 Punkte, wenn das Programm korrekt funktioniert. Eine `aufgabe1_test.cpp` sagt ihnen, ob das der Fall ist (verfügbar ab dem Tutorium).

Es gibt 4 Zusatzpunkte wenn ihr Programm auch mit '?' (Wildcards) sowohl im Text, als auch in der Query umgehen kann. Eine Wildcard kann dabei jedes andere Zeichen des Alphabets matchen.

Weitere Hinweise: Einloggen in einen Linux Rechner an der Uni:

Windows-Nutzer können das Tool putty.exe (www.putty.org) benutzen um sich auf einen Linux-Pool Rechner einzuloggen.

Linux oder Mac Nutzer können sich einfach per ssh mit einem Linux-Pool Rechner einloggen.

Beides benötigt einen VPN-Client, um sich in das FU Netzwerk einzuwählen, falls sie Ihren privaten Internetanschluss benutzen. Aus dem FU Netzwerk (z.B. Eduroam) können Sie sich direkt auf den Poolrechnern einloggen (ohne VPN). Alternativ zum VPN können Sie auch einen der Remote Login Server benutzen (http://www.mi.fu-berlin.de/w/IT/ComputeServer#Remote_Login_Server)

Folgende Poolrechner bzw ComputeServer stehen Ihnen zur Verfügung:

<http://www.mi.fu-berlin.de/w/IT/ServicesStudentPools>

http://www.mi.fu-berlin.de/w/IT/ComputeServer#Linux_standalone

Viel Erfolg!