

Prof. Dr. Christoph Scholl
Dr. Tim Welschhold
Alexander Konrad
Niklas Wetzel

Freiburg, 02.12.2022

Betriebssysteme

Musterlösung zu Übungsblatt 7

Aufgabe 1 (3 + 8 Punkte)

Der Euklidische Algorithmus berechnet den größten gemeinsamen Teiler zweier natürlicher Zahlen:

„Euklid berechnete den größten gemeinsamen Teiler, indem er nach einem gemeinsamen Maß für die Längen zweier Linien suchte. Dazu zog er wiederholt die kleinere der beiden Längen von der größeren ab. Dabei nutzt er aus, dass sich der größte gemeinsame Teiler zweier Zahlen (oder Längen) nicht ändert, wenn man die kleinere von der größeren abzieht.“ (Quelle: wikipedia)

- a) Schreiben Sie ein C-Programm das den Euklidischen Algorithmus **rekursiv** implementiert. Verwenden Sie dafür die folgende Schablone

```
int function ggt(int a, int b)
{
    //...
}

void main()
{
    int ggt_result;
    ggt_result = ggt(16, 12); //Ruecksprungadresse 100
};
```

- b) Geben Sie den Inhalt des Stacks (und von ACC¹, falls interessant) genau vor den Sprüngen zum Code für die ggT-Funktion `ggt(a,b)` und genau nach den Rücksprüngen aus dem Code für die ggT-Funktion an. Nehmen Sie an, dass die erste freie Stack-Speicherzelle die Ihnen zur Verfügung steht mit 400 adressiert wird. Eine Rücksprungadresse ist bereits im Kommentar festlegt, weitere Rücksprungadressen dürfen Sie selbst festlegen.

Lösung:

¹Wie in der Vorlesung nehmen wir an, dass das Ergebnis im ACC zurückgegeben wird.

a)

```
int function ggt(int a, int b)
{
    if(a == b)
        return a;
    else if(a < b)
        return ggt(a, b-a); //Ruecksprungadresse 200
    else if(a > b)
        return ggt(a-b, b); //Ruecksprungadresse 300 -exakt anhand von 200
                                // berechnen koennen wir noch nicht
}

void main()
{
    int ggt_result;
    ggt_result = ggt(16, 12); //Ruecksprungadresse 100
};
```

b) Ablauf: ggt(16,12) ggt(4,12) ggt(4,8) ggt(4,4) Ergebnis 4

		lokale Stackinhalte
		lokale Variablen
		Formale Parameter
		Ruecksprungadresse
		Beginn Vorgängerframe
BAF- >		
397	12	
398	16	
399	100	
400	egal	
393	12	
394	4	
395	300	
396	400	
397	12	
398	16	
399	100	
400	egal	
389	8	
390	4	
391	200	
392	396	
393	12	
394	4	
395	300	
396	400	
397	12	
398	16	
399	100	
400	egal	

385	4
386	4
387	200
388	392
389	8
390	4
391	200
392	396
393	12
394	4
395	300
396	400
397	12
398	16
399	100
400	egal

nach jedem Stackabbau steht in ACC Rückgabewert 4 [1] für korrekte Angabe vom ACC Inhalt

Stack abbauen, dann wird **ggt_result** auf 4 gesetzt [1] für richtiges Ergebnis. [1] pro richtigem Stack Zustand, insgesamt [7] (nach dem 4. Zustand werden die vorherigen Zustände rückwärts abgelaufen, wenn die einzelnen Stackframes nacheinander abgebaut werden).

Aufgabe 2 (1 + 2 + 2 Punkte)

Dateirechte werden unter Linux mit dem Befehl `chmod` gesetzt. Informieren Sie sich mit `man chmod` über die Verwendung des Befehls.

Geben Sie für die Aufgabenteile b) und c) in Ihrer Abgabe die ausgeführten Befehle und deren Ausgabe an.

- a) Nehmen Sie an, Sie führen den Befehl `ls -l` aus und erhalten dabei folgendes Ergebnis:

```
$ ls -l
drwxr-x--x 2 ak1062 uni      26 27. Okt 14:06 meine_dateien
```

Erläutern Sie, was diese Zugriffsrechte im Falle eines Verzeichnisses bedeuten.

- b) Erstellen Sie im Ordner `/tmp` einen Unterordner mit Ihrem Benutzernamen und kopieren Sie das Systemprogramm `whoami` mit folgenden Befehlen (Stellen Sie durch Aufruf von `bash` am Anfang sicher, dass Sie mit der `bash`-Shell arbeiten):

```
cd /tmp
mkdir $(whoami)
cd $(whoami)
cp /usr/bin/whoami werbinich
```

Sie können das kopierte Programm dann mit „`./werbinich`“ ausführen.

- 1) Vergleichen Sie mithilfe von „`ls -l`“ die Eigenschaften des Originalprogramms „`/usr/bin/whoami`“ mit Ihrer Kopie „`werbinich`“. Welche Eigenschaften haben sich geändert?
 - 2) Ändern Sie bei Bedarf die Attribute der Datei „`werbinich`“ so, dass alle Mitglieder der Gruppe `uni` die Datei ausführen können. Falls diese Gruppe noch nicht existiert, müssen sie diese zunächst erstellen und Mitglied der Gruppe werden. Welcher Username wird einem Nutzer `xy1234` (der Mitglied der Gruppe `uni` ist) angezeigt, wenn er „`./werbinich`“ ausführt?
 - 3) Wie können Sie erreichen, dass ein anderer Nutzer (aus der Gruppe `uni`) bei der Ausführung nicht seinen Login, sondern den Login des Besitzers der Datei als Ausgabe erhält?
Hinweis: `whoami` gibt die Benutzerkennung aus, unter der das Programm läuft.
- c) Wechseln Sie mit `cd` in Ihr Home-Verzeichnis (`/home/benutzername`). Legen Sie dort mit dem Befehl `mkdir systeme-public` ein neues Unterverzeichnis an.
- 1) Setzen Sie die Rechte von `systeme-public` so, dass jeder Benutzer lesend auf den Inhalt zugreifen und in den Ordner navigieren kann.
 - 2) Was müssen Sie nun tun, damit immer noch jeder auf `systeme-public` zugreifen kann, aber aus Gründen der Sicherheit der Inhalt des übergeordneten Homes für alle Benutzer außer dem Besitzer nicht mehr aufgelistet werden kann?

Bei Benutzer **konrad** würde dies so aussehen: Der Inhalt von `/home/konrad` ist für andere Benutzer nicht sichtbar, ein `ls /home/konrad/systeme-public` zeigt aber trotzdem alle vorhandenen Dateien des Unterordners an.

Tipps zur Ausführung: Sie können diese Aufgabe auf einer eigenen Maschine mit Linux-Betriebssystem, z.B. Ubuntu ausführen oder ein Linux-Betriebssystem über VirtualBox an Ihrem vorhandenen System einrichten. Hinweise und Links zu Ubuntu und VirtualBox finden Sie auf Übungsblatt 0. Alternativ kann die Aufgabe auch wie zuvor bei Übungsblatt 1 auf dem Login-Rechner der Universität (Verbindung per ssh mit `ssh xy1234@login.uni-freiburg.de`) bearbeitet werden.

Lösung:

a) Dateirechte [1]:

- Der Besitzer **ak1062** darf den Verzeichnisinhalt auflisten (r), Dateien erstellen, löschen und umbenennen (w) und in das Verzeichnis wechseln (x).
- Mitglieder der Gruppe **uni** dürfen nur den Verzeichnisinhalt auflisten und in das Verzeichnis wechseln.
- Alle anderen Benutzer dürfen nur in das Verzeichnis wechseln, den Inhalt aber nicht auflisten.

b) 1) Beim Kopieren von `/usr/bin/whoami` nach **werbinich** ändert sich der Besitzer und die Gruppe der Datei von `root:root` nach `<benutzername>:student` [0.5]. Die Zugriffsrechte können erhalten bleiben oder teilweise verloren gehen.

(Zur Erklärung: Falls Zugriffsrechte verloren gehen, liegt das daran, dass die Zugriffsrechte mit der `umask` maskiert werden.

Beispiele:

`umask 0002` beim Kopieren wird Schreibrecht ($w = 2$) für Others gelöscht.

`umask 0077` beim Kopieren werden alle Rechte ($r+w+x = 4+2+1 = 7$) für Gruppe und Others gelöscht.

Die führende 0 gibt an, dass es sich um Oktaldarstellung handelt.

Mit `umask -S` lassen sich die Rechte von neu erstellen Dateien anzeigen.

`umask` wurde in der Vorlesung nicht besprochen.)

2) `chmod g=rx werbinich` [0.5]. Gegenfalls muss zunächst die Gruppe der Datei mit `chgrp uni werbinich` geändert werden. Es existieren mehrere Lösungsvarianten, u.U. müssen keine Rechteänderungen vorgenommen werden. Falls die Gruppe **uni** noch nicht existiert [keine Punkte]:

- Gruppe erstellen: `sudo groupadd uni`.
- Mitglied der Gruppe werden `sudo usermod -a -G uni $USER`.
- Ausloggen und wieder einloggen, um Gruppenmitgliedschaft zu erlangen.
- Späteres Löschen der Gruppe mit `sudo groupdel uni`.

3) Das Programm **werbinich** zeigt den Namen des Nutzers (**xy1234**) an, da das Programm unter seiner Benutzerkennung ausgeführt wird. Um das Programm stattdessen unter der Benutzerkennung des Dateieinhabers auszuführen, muss das SUID-Bit (Set User ID) gesetzt werden: [1]

```
[chgrp uni werbinich]
chmod g=rx werbinich
chmod u+s werbinich
./werbinich
```

c) Rechtevergabe [je 1 pro chmod]

```
cd
mkdir systeme-public
1) chmod go=rx systeme-public
2) chmod go=x ~
```

Das `x`-Recht muss für alle übergeordneten Verzeichnisse gesetzt sein, da der Gruppenpartner sonst nicht mit `cd` in `systeme-public` wechseln darf (selbst wenn `x` auf `systeme-public` selbst gesetzt ist). Wenn der Gruppenpartner in der selben Gruppe (`student`) ist, reichen auch Gruppenrechte aus.

Abgabe: als PDF im Übungsportal bis 09.12.2022 um 12:00