



Betriebssysteme und Sicherheit, WS 2021/22

2. Aufgabenblatt – Unix II

Bearbeitungszeitraum: 25.10.2021 – 29.10.2021

Ziel dieses Übungsblattes ist es diverse Grundlagen von Unix und der systemnahen Programmierung zu erlernen und zu festigen. Dabei werden wir uns erst beispielhaft mit der Entwicklung von C-Programmen befassen, und damit, wie typischerweise Programme den Hauptspeicher verwenden. Als zweiten Themenkomplex werden wir uns dann mit der Interaktion von Programmen beschäftigen – mit dem Nutzer über Ein- und Ausgabe, und mit anderen Programmen über *Pipes* oder *Dateien*.

Aufgabe 2.1 Erläutern Sie die einzelnen Schritte des Übergangs von einem (gedanklich oder schriftlich vorliegenden) Algorithmus über die Implementierung in C bis hin zu einem lauffähigen Programm. Geben Sie dazu jeweils folgende Informationen an:

- Schritt bzw. ausführendes Programm
- Programmname in Unix
- Ergebnis (Bezeichnung und Art der erzeugten Datei, Beschreibung ihres Inhalts)
- Dateiendung unter Unix

Aufgabe 2.2 In der Vorlesung wurde die Adressraumstruktur von Unix eingeführt. Ordnen Sie für das links stehende Beispiel die in der Tabelle enthaltenen Symbole demgemäß ein.

<pre>#include <stdlib.h> int a[20]; int x = 1; void foo(void) { int b[20]; void *p = malloc(100); free(p); }</pre>	<pre>a x foo b p *p</pre>	<table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="border: none;"></th> <th style="border: none;">Text</th> <th style="border: none;">Data</th> <th style="border: none;">BSS</th> <th style="border: none;">Heap</th> <th style="border: none;">Stack</th> </tr> </thead> <tbody> <tr> <td style="border: none;">a</td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> </tr> <tr> <td style="border: none;">x</td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> </tr> <tr> <td style="border: none;">foo</td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> </tr> <tr> <td style="border: none;">b</td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> </tr> <tr> <td style="border: none;">p</td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> </tr> <tr> <td style="border: none;">*p</td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> <td style="border: none;"><input type="checkbox"/></td> </tr> </tbody> </table>		Text	Data	BSS	Heap	Stack	a	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	foo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	b	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	p	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	*p	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Text	Data	BSS	Heap	Stack																																							
a	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																							
x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																							
foo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																							
b	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																							
p	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																							
*p	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																							

Aufgabe 2.3 In Unix hat jedes Programm standardmäßig drei Dateien geöffnet: `stdin`, `stdout` und `stderr`. Weshalb ist ein separater Kanal für Fehlermeldungen (`stderr`) sinnvoll, wenn doch bereits `stdout` existiert und beide zur Ausgabe verwendet werden?

Aufgabe 2.4 Welche Voraussetzungen muss ein Programm erfüllen, das in einer Kommandosequenz wie `ls | grep ps | sort` verwendet werden soll, oder anders gefragt, warum kann das Beispiel `latex foo.tex | dvips | lpr` so nicht funktionieren? Informieren Sie sich zuvor mit Hilfe der Manpages über die Funktionalität der auftretenden Kommandos.

Aufgabe 2.5 Unix bietet die Systemaufrufe `creat(2)`, `unlink(2)`, `write(2)` und `close(2)` respektive zum Anlegen/Öffnen, Löschen, Schreiben und Schließen einer Datei an (Details siehe zugehörige Manpages). Was geschieht beim Ausführen der folgenden Codesequenz?

```
int fd = creat("/tmp/test", S_IRWXU);
unlink("/tmp/test");
write(fd, "Hello_world\n", 12);
close(fd);
```

Wann wird die Datei `test` gelöscht und ist damit für andere nicht mehr sichtbar? Kann `write()` erfolgreich abgeschlossen werden? Was geschieht beim Aufruf von `close()`?