

Übungen zu Systemnahe Programmierung in C

Abschnitt 9.2: Hands-On (Laufschrift)

22.06.2020

Tim Rheinfels
Benedict Herzog
Bernhard Heinloth

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg



Lehrstuhl für Verteilte Systeme
und Betriebssysteme



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT



- Funktionsweise:
Schrittweises Anzeigen eines Textes auf der 7-Segment-Anzeige
- Lernziele:
 - Zeichenketten in C
 - Zeiger & Zeigerarithmetik
 - Alarmer & Schlafenlegen
- Vorgehen:
 - Wiederkehrender Alarm mittels `TIMER0`
 - Zusammensetzen des aktuellen Teilstrings
 - Ausgabe über 7-Segment-Anzeige
 - In Wartephase Mikrocontroller in den Energiesparmodus versetzen (Passives Warten)



```
01 const char *string = "HALLO SPIC";
02 const char *current = string;
03 // current[0] == 'H' && current[1] == 'A'
04 ++current;
05 // current[0] == 'A' && current[1] == 'L'
06 // [...]
07 // current[0] == '\0', current[1] == ??
08 current = string;
```



```
01 const char *string = "HALLO SPIC";
02 const char *current = string;
03 // current[0] == 'H' && current[1] == 'A'
04 ++current;
05 // current[0] == 'A' && current[1] == 'L'
06 // [...]
07 // current[0] == '\0', current[1] == ??
08 current = string;
```



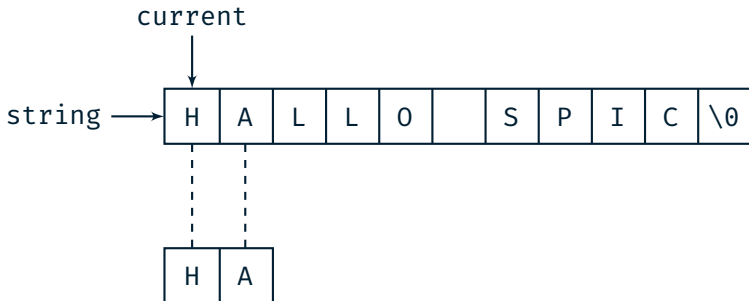


```
01 const char *string = "HALLO SPIC";  
02 const char *current = string;  
03 // current[0] == 'H' && current[1] == 'A'  
04 ++current;  
05 // current[0] == 'A' && current[1] == 'L'  
06 // [...]  
07 // current[0] == '\0', current[1] == ??  
08 current = string;
```



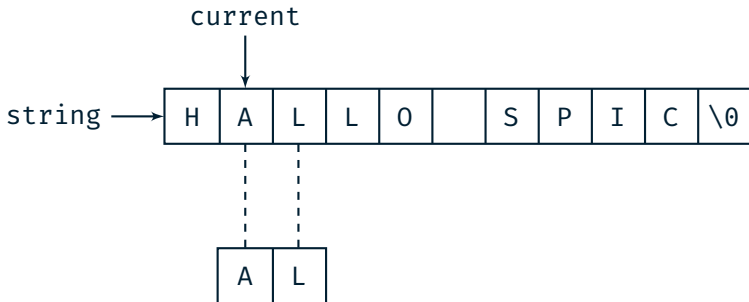


```
01 const char *string = "HALLO SPIC";  
02 const char *current = string;  
03 // current[0] == 'H' && current[1] == 'A'  
04 ++current;  
05 // current[0] == 'A' && current[1] == 'L'  
06 // [...]  
07 // current[0] == '\\0', current[1] == ??  
08 current = string;
```





```
01 const char *string = "HALLO SPIC";  
02 const char *current = string;  
03 // current[0] == 'H' && current[1] == 'A'  
04 ++current;  
05 // current[0] == 'A' && current[1] == 'L'  
06 // [...]  
07 // current[0] == '\\0', current[1] == ??  
08 current = string;
```





```
01 const char *string = "HALLO SPIC";
02 const char *current = string;
03 // current[0] == 'H' && current[1] == 'A'
04 ++current;
05 // current[0] == 'A' && current[1] == 'L'
06 // [...]
07 // current[0] == '\0', current[1] == ??
08 current = string;
```

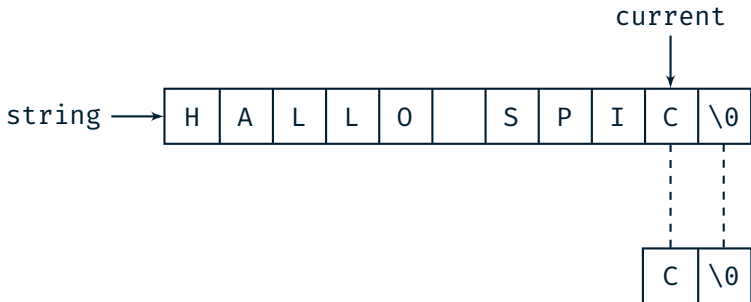
string →

H	A	L	L	O		S	P	I	C	\0
---	---	---	---	---	--	---	---	---	---	----

...

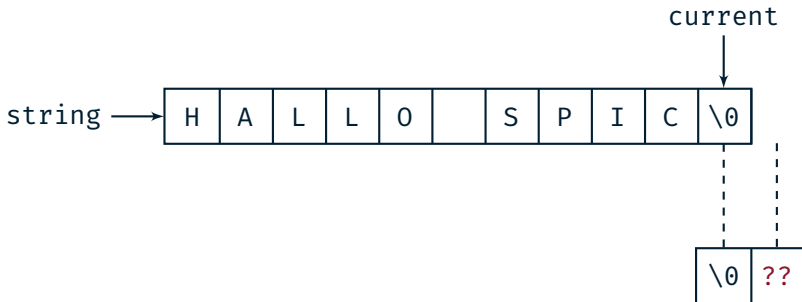


```
01 const char *string = "HALLO SPIC";
02 const char *current = string;
03 // current[0] == 'H' && current[1] == 'A'
04 ++current;
05 // current[0] == 'A' && current[1] == 'L'
06 // [...]
07 // current[0] == '\0', current[1] == ??
08 current = string;
```



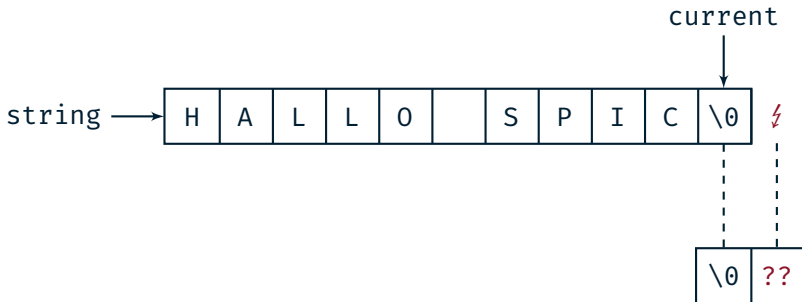


```
01 const char *string = "HALLO SPIC";
02 const char *current = string;
03 // current[0] == 'H' && current[1] == 'A'
04 ++current;
05 // current[0] == 'A' && current[1] == 'L'
06 // [...]
07 // current[0] == '\0', current[1] == ??
08 current = string;
```





```
01 const char *string = "HALLO SPIC";
02 const char *current = string;
03 // current[0] == 'H' && current[1] == 'A'
04 ++current;
05 // current[0] == 'A' && current[1] == 'L'
06 // [...]
07 // current[0] == '\0', current[1] == ?? ⚡
08 current = string;
```





```
01 const char *string = "HALLO SPIC";  
02 const char *current = string;  
03 // current[0] == 'H' && current[1] == 'A'  
04 ++current;  
05 // current[0] == 'A' && current[1] == 'L'  
06 // [...]  
07 // current[0] == '\0', current[1] == ?? ⚡  
08 current = string;
```

