

System-Programmierung (syspr) 08. November 2022

thomas.amberg@fhnw.ch

Assessment I

Vorname:	<u> </u>	Punkte:	_ / 90,	Note:
Name:		Frei lassen	für Korrek	tur.
Klasse:	3ib			
Hilfsmitte	el:			
- Ein A4-I	Blatt handgeschriebene Zusammenfassung.			
- Lösen Si	e die Aufgaben jeweils direkt auf den Prüfungs	blättern.		
- Zusatzbl	ätter, falls nötig, mit Ihrem Namen und Frage	n-Nr. auf jede	em Blatt.	
Nicht erla	ubt:			
- Unterlag	gen (Slides, Bücher,).			
- Comput	er (Laptop, Smartphone,).			
- Kommu	nikation mit anderen Personen.			
Bewertun	g:			
- Multiple	Response: \square <i>Ja</i> oder \square <i>Nein</i> ankreuzen, +1/-	-1 Punkt pro	richtige/fal	sche Antwort,
beide ni	cht ankreuzen ergibt +0 Punkte; Total pro Fraș	ge gibt es nie	weniger als	s 0 Punkte.
- Offene F	ragen: Bewertet wird Korrektheit, Vollständigl	keit und Kürz	ze der Antw	ort.
- Program	nme: Bewertet wird die Idee/Skizze und Umset	zung des Pro	gramms.	
Fragen zu	r Prüfung:			
- Währen	d der Prüfung werden vom Dozent keine Frage	n zur Prüfun	g beantwor	tet.

- Ist etwas unklar, machen Sie eine Annahme und notieren Sie diese auf der Prüfung.

Erste Schritte in C

1) V	<i>N</i> elche	dieser	Deklarat	ionen von	main()	sind k	korrekt in (C3
------	----------------	--------	----------	-----------	--------	--------	--------------	----

Punkte: _ / 4

Zutreffendes ankreuzen

2) Welche Abfolge von Statements führt zu folgender Situation im Speicher? Punkte: _ / 4



Zutreffendes ankreuzen

```
□ Ja | □ Nein int i = 3; int *p = &i; int **q = &p; int *r = p; 
□ Ja | □ Nein int i = 3; int *p = &i; int *r = *p; int *q = &p; 
□ Ja | □ Nein int i = 3; int *p = &i; int *q = p; int *r = &i; 
□ Ja | □ Nein int i = 3; int *r = &i; int *p = &i; int **q = &p;
```

3) Gegeben die folgenden Structs, welche Statements sind wahr?

Punkte: _ / 4

```
struct r { long a; long b; long c; };
struct s { long n; struct r *p; };
```

Zutreffendes ankreuzen

□ Ja □ Nein	<pre>sizeof(struct s) > sizeof(struct r)</pre>
□ Ja □ Nein	<pre>sizeof(struct r) == 3 * sizeof(long)</pre>
□ Ja □ Nein	<pre>sizeof(struct s) == sizeof(long) + sizeof(struct r)</pre>
□ Ja □ Nein	<pre>sizeof(struct r *) < 2 * sizeof(long)</pre>

Funktionen in C

4) Wieso führt das folgende Programm zu einem Segmentation Fault? Punkte: _ / 6

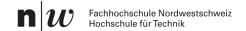
```
01: void f(int (*g)(void)) {
02:    int i = g();
03: }
04:
05: int g(void) {
06:    return 0;
07: }
08:
09: int main(void) {
10:    f(main);
11:    return 0;
12: }
```

Erklärung hier eintragen, Schritt für Schritt ausformulieren:

5) Wie funktioniert ein System Call?

Punkte: _ / 4

 $Erkl\"{a}rung\ hier\ eintragen,\ vier\ wesentliche\ Punkte\ ausformulieren:$



6) Schreiben Sie ein Programm <i>max</i>	, welches das längste der	<i>n</i> übergebenen Aı	rgumente, ode	r
wie ganz unten, das erste von mehre	ren maximal langen Arg	umenten, ausgibt.	Punkte: / 1	2

```
$ ./max
$ ./max short looong
looong
$ ./max short words use them
short
```

Hier ein Auszug aus der Doku, #includes und Fehlerbehandlung können Sie weglassen:

```
int printf(const char *format, ...); // format string %s, char %c, int %d
size_t strlen(const char *s); // calculate the length of a string
```

Idee (kurz) und Source Code hier, oder auf Zusatzblatt mit Ihrem Namen und Fragen-Nr.:

File In-/Output

7) Schreiben Sie ein Programm patch, das ein existierendes File orig an der Position n mit Bytes des existierenden Files delta der Länge m überschreibt, wie hier gezeigt. Punkte: _ / 12

```
$ echo "hello?" > orig
$ echo "ipp" > delta
$ ./patch orig 1 delta 3
$ cat orig
hippo?
```

Verwenden Sie dazu die folgenden System Calls, Fehlerbehandlung können Sie weglassen:

```
int atoi(const char *nptr); // convert a string to an integer

off_t lseek(int fd, off_t offset, int from); // Position read/write file
offset; from = SEEK_SET, SEEK_CUR or SEEK_END; returns new offset from 0.

int open(const char *pathname, int flags); // Opens the file specified by
pathname. Returns the file descriptor. Flags include O_RDONLY, O_WRONLY.

ssize_t read(int fd, void *buf, size_t n); // Attempts to read up to n
bytes from file descriptor fd into buf. Returns number of bytes read ≤ n.

ssize_t write(int fd, const void *buf, size_t n); // Writes up to n bytes
from buf to the file referred to by fd. Returns nr. of bytes written ≤ n.
```

Idee (kurz) und Source Code hier, oder auf Zusatzblatt mit Ihrem Namen & Fragen-Nr.:

8) Schreiben Sie ein Programm fdb, das ein File db als "Array DB" verwaltet, für Strings < 32 Byte, die mit put und get am Index i schreib- bzw. lesbar sind, wie hier gezeigt. Punkte: $_/$ 12

```
$ ./fdb my.db put 3 hello
$ ./fdb my.db get 3
hello
```

Verwenden Sie dazu die folgenden System Calls, Fehlerbehandlung können Sie weglassen:

int atoi (const char *nptr); // convert a string to an integer
off_t lseek(int fd, off_t offset, int from); // Position read/write file offset; from = SEEK_SET, SEEK_CUR or SEEK_END; returns new offset from 0. Allows file offset to be set beyond the end of the file, fills with '\0'
int open (const char *pathname, int flags, mode_t mode); // Opens the file specified by pathname. Or creates it if O_CREAT is used. Returns the file descriptor. Flags include O_APPEND, O_CREAT, O_RDWR, O_RDONLY, O_WRONLY. Modes, which are used together with O_CREAT include S_IRUSR and S_IWUSR
ssize_t read (int fd, void *buf, size_t n); // Attempts to read up to n bytes from file descriptor fd into buf. Returns number of bytes read ≤ n
ssize_t write (int fd, const void *buf, size_t n); // Writes up to n bytes from buf to the file referred to by fd. Returns nr. of bytes written ≤ n
int printf (const char *format,); // format string %s, char %c, int %d
size_t strlen (const char *s); // calculate length of a string, excl. '\0'

Idee (kurz) und Source Code hier, oder auf Zusatzblatt mit Ihrem Namen & Fragen-Nr.:

Prozesse und Signale

9) Schreiben Sie ein Programm count, das ausgibt, wie weit ein Prozess in 1 Sekunde zählen kann. Nutzen Sie dazu die alarm() Funktion, die das SIGALRM Signal auslöst. Punkte: $_/$ 12

```
$ ./count
counted to 75654776
```

Verwenden Sie dazu die folgenden System Calls, Fehlerbehandlung können Sie weglassen:

```
int alarm(int sec); // arranges for a SIGALRM signal to be delivered to
the calling process in sec seconds; returns remaining sec of prev. alarm
int pause(void); // sleep until a signal causes invocation of a handler

typedef void (*sighandler_t)(int); e.g. SIGINT = 2, ^C; SIGTSTP = 20, ^Z
sighandler_t signal(int sig, sighandler_t handler); // set SIG_IGN,
SIG_DFL, or a programmer-defined function to handle the signal sig
int printf(const char *format, ...); // format string %s, char %c, int %d
```

Idee (kurz) und Source Code hier, oder auf Zusatzblatt mit Ihrem Namen und Fragen-Nr.:

Prozess Lebenszyklus

10) Schreiben Sie ein Programm exec, das ein per Command Line gegebenes Programm in n parallelen Child Prozessen ausführt, wie hier gezeigt, und auf deren Ende wartet. P.kte: $_/$ 12

```
$ ./exec 3 hello
Hello, World!
Hello, World!
Hello, World!
```

Verwenden Sie dazu die folgenden System Calls, Fehlerbehandlung können Sie weglassen:

```
int atoi(const char *nptr); // convert a string to an integer
pid_t fork(void); // create a child process, returns 0 in child process
int execve(const char *pathname, char *const argv[], char *const envp[]);
// executes the program referred to by pathname; argv, envp can be NULL
void exit(int status); // cause normal process termination
pid_t wait(int *wstatus); // wait for child process to terminate; returns the process ID of the terminated child or -1 if no child left to wait for
```

Idee (kurz) und Source Code hier, oder auf Zusatzblatt mit Ihrem Namen & Fragen-Nr.:

void *start(void *arg) {

printf("A\n");

printf("B\n");
pthread_t thread;

printf("C\n");

printf("D\n");

pthread_mutex_lock(&m);

pthread_cond_signal(&c);
pthread_mutex_unlock(&m);

pthread_mutex_lock(&m);

pthread_cond_wait(&c, &m);

pthread_mutex_unlock(&m);

Threads

}

}

int main() {

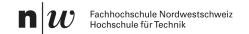
11) Was ist der Output dieses Programms, und wieso?

pthread_mutex_t m = PTHREAD_MUTEX_INITIALIZER; pthread_cond_t c = PTHREAD_COND_INITIALIZER;

pthread_create(&thread, NULL, start, NULL);

```
Punkte: _ / 8
```

Output und Begründung hier eintragen; Annahme: #includes sind vorhanden:



Zusatzblatt zu Aufgabe Nr	von (Name)	