

Exercises for Memory Leaks and Debugging with Valgrind

Jula Menck

2014-06-19

1.)

Install Valgrind on a compatible machine and play around with it for a bit with some code and different options to get used to it. If you have no code lying around use the following. This code is in working condition and should give you no error messages with Memcheck. Comment out some lines and recompile to see how the error messages change:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     char *p = malloc(1);
7     *p = 'a';
8     char c = *p;
9
10    printf("\n [%c]\n", c);
11    free (p);
12    return 0;
13 }
```

2.)

Compile the following code and use Memcheck on it to see the error. Correct the error so that Memcheck will no longer give an error message:

```
1 #include <stdlib.h>
2
3 int main(void)
4 {
5     int *ip;
6     ip=(int *) malloc(sizeof(int));
7     *ip=4711;
8     ip=(void *)0;
9     free(ip);
10 }
```

3.)

a) Compile the following code and use Memcheck on it to see the error. Correct the error, so that Memcheck will no longer give an error message:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     char *p = malloc(1);
7     *p = 'a';
8
9     char c = *p;
10
11     printf("\n [%c]\n",c);
12
13     free(p);
14     c = *p;
15     return 0;
16 }
```

b) Do the same as in a) with this code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     char *p = malloc(1);
7     *p = 'a';
8
9     char c = *(p+1);
10
11     printf("\n [%c]\n",c);
12
13     free(p);
14     return 0;
15 }
```