Advanced Programming in the UNIX Environment

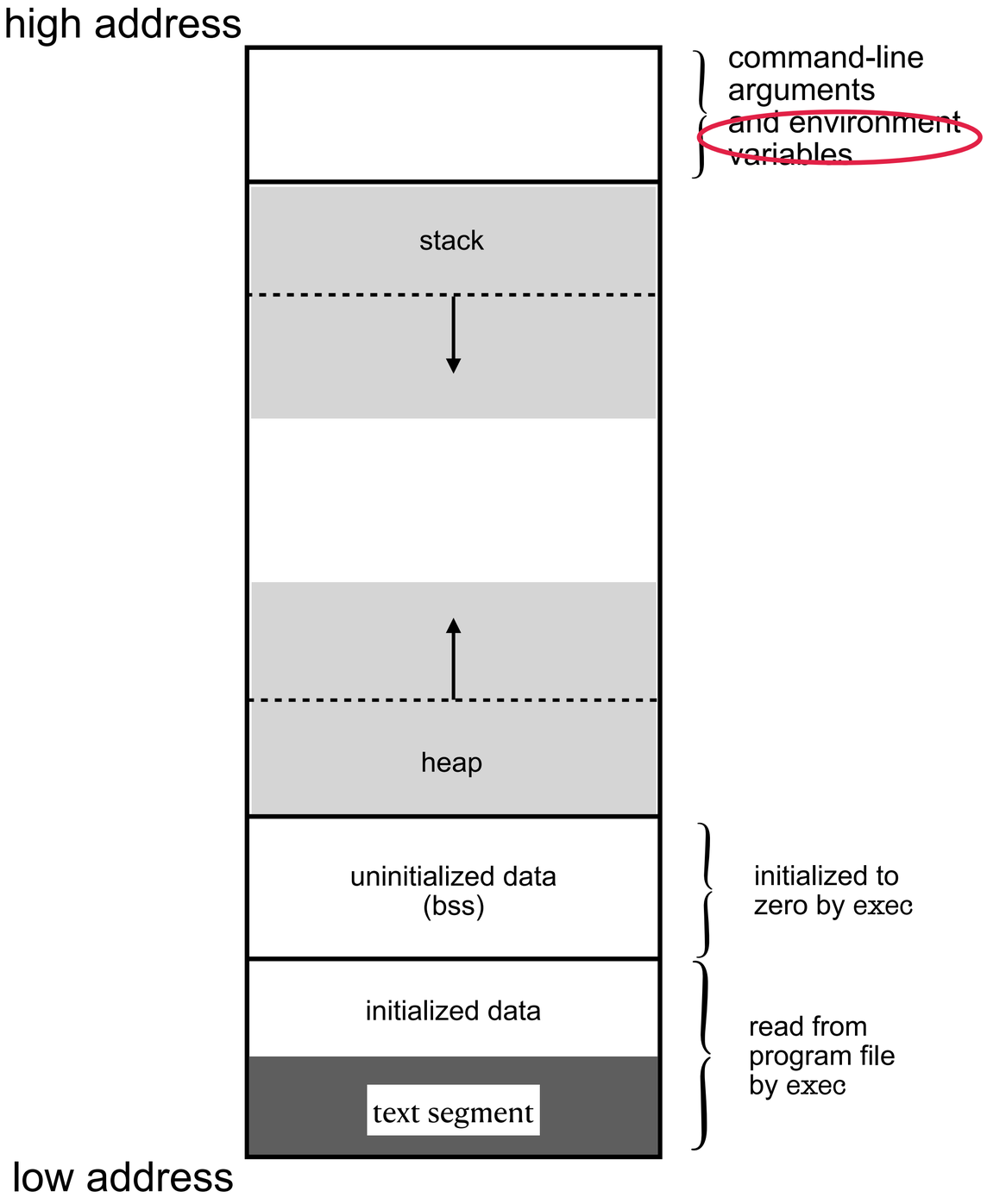
Week 06, Segment 4:
The Environment

Department of Computer Science
Stevens Institute of Technology

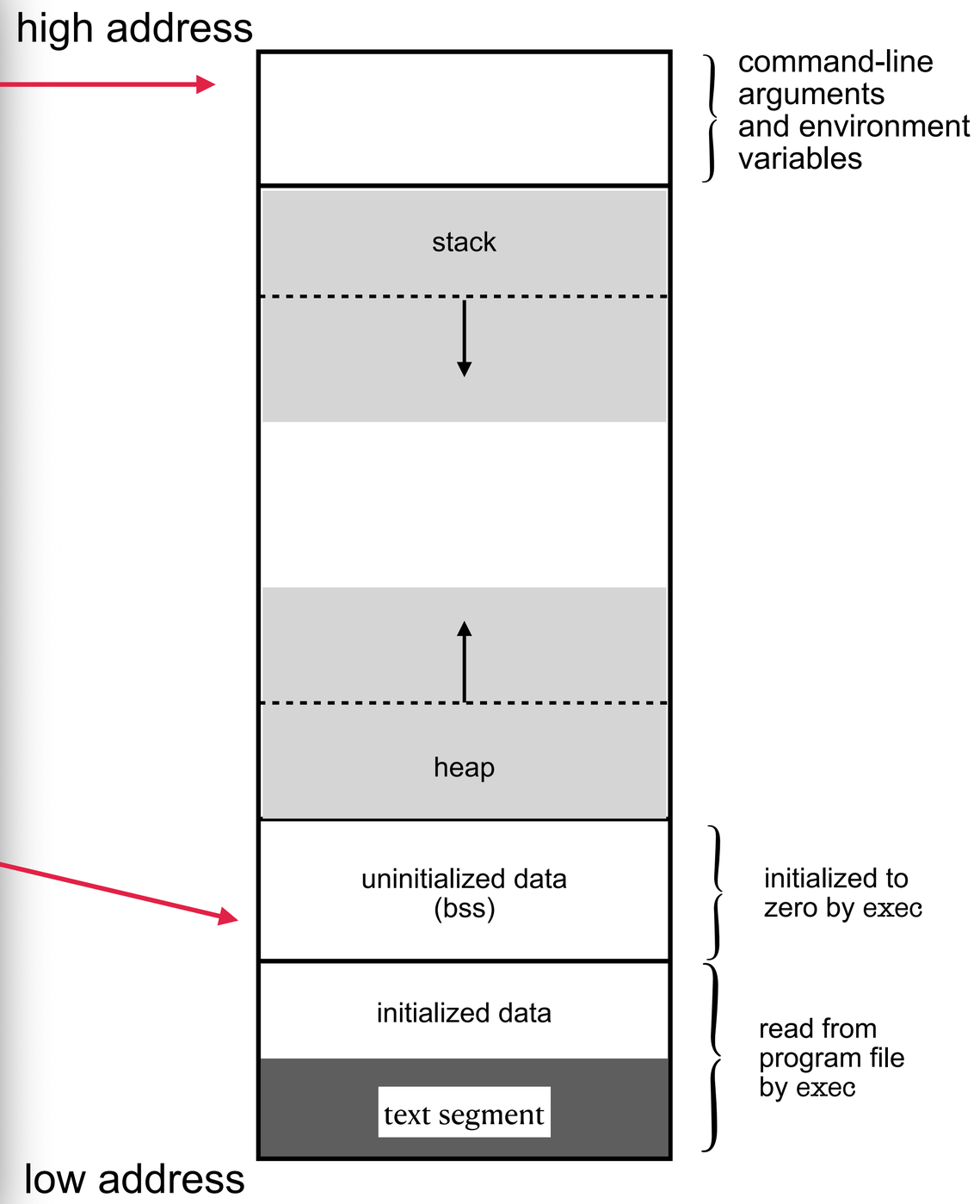
Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/631/


Terminal — 80×24
apue$ env
ENV=/home/jschauma/.shrc
PS1=apue$
PWD=/home/jschauma/06
MAIL=/var/mail/jschauma
HOME=/home/jschauma
PATH=/home/jschauma/bin:/bin:/sbin:/usr/bin:/usr/sbin:/usr/X11R7/bin:/usr/pkg/bi
n:/usr/pkg/sbin:/usr/games:/usr/local/bin:/usr/local/sbin
SSH_CONNECTION=10.0.2.2 61988 10.0.2.15 22
SSH_TTY=/dev/pts/1
TERM=xterm-256color
OLDPWD=/home/jschauma
USER=jschauma
SSH_AUTH_INFO_0=publickey ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCv0TegkoPwNoeNf8
yWpc+N5Q2SS8Gz0pwQ88xsjElrPl4KsulCOn3EGu8Ry/SJdV8CcnfgOWt21vMRajltC3sxxmCKjLOkAR
pPk++EVPu/WbkP5tsuKBPAPlaKUyfGiYkQtaPW04eP5Crr+c0qbyAqn+cHnAP/sMFuZvHPwhFK3ywaR/
K/o87hrlHg030rn7v4wDEn/a+TJMmpLZ12IQO6/PcfFuyKySuf+NXd/3m4dX9UPIzH5g4Y4HrEVjK4aE
Au9kAsd1SfkyUZUoRat0NAEanUm7hznzffJRrdMs1s2pixbOBdSUOQNy2E67Iwa7Oyf2fxMQDorQJFxJ
VuC5sN

CFLAGS=-Wall -Werror -Wextra
SSH_CLIENT=10.0.2.2 61988 22
EDITOR=vi
LOGNAME=jschauma


high address
command-line arguments and environment variables
stack
heap
uninitialized data (bss)
initialized to zero by exec
initialized data
text segment
read from program file by exec
low address

```
Terminal — 80×44
apue$ ./a.out
High address (args and env):
----------------------------
environ[0] at                                          : 0x7F7FFF8A82B8
last arg at                                            : 0x7F7FFF8A82B0
first arg at                                           : 0x7F7FFF8A82A8

Stack:
------
First variable inside main at                          : 0x7F7FFF8A8254
func_array[] ends at                                   : 0x7F7FFF8A8250
func_array[] (like 'array[]', but on stack) begins at  : 0x7F7FFF8A8240
argc at                                                : 0x7F7FFF8A823C
argv at                                                : 0x7F7FFF8A8230
func2 (from main): frame at                            : 0x7F7FFF8A821C
func frame at                                          : 0x7F7FFF8A821C
static int n within func at                            : 0x      601A0C
func2 (from func): frame at                            : 0x7F7FFF8A81FC

Heap:
-----
malloced area ends at                                  : 0x76AC2C04B020
malloced area begins at                                : 0x76AC2C04B000

Uninitialized Data (BSS):
-------------------------
array[] ends at                                        : 0x      601AB0
array[] (uninitialized, fixed-size char * on BSS) from : 0x      601AA0
num2 (uninitialized global int) at                     : 0x      601A98
string2 (uninitialized global char *) at               : 0x      601A90
extern **environ at                                    : 0x      601A80

Initialized Data:
-----------------
num (initialized global int) at                        : 0x      601A08
string (initialized global char *) at                  : 0x      601A00

Text Segment:
-------------
func2 (function) at                                    : 0x      400E53
func (function) at                                     : 0x      400E02
main (function) at                                     : 0x      400B1A

apue$
```



Terminal — 80×24

```
        relocate_self(ps_strings);
#endif

        if (ps_strings == NULL)
                _FATAL("ps_strings missing\n");
        __ps_strings = ps_strings;

        environ = ps_strings->ps_envstr;

        if (ps_strings->ps_argvstr[0] != NULL) {
                char *c;
                __progname = ps_strings->ps_argvstr[0];
                for (c = ps_strings->ps_argvstr[0]; *c; ++c) {
                        if (*c == '/')
                                __progname = c + 1;
                }
        } else {
                __progname = empty_string;
        }
+-- 29 lines: if (cleanup != NULL)----------------------------------------

        exit(main(ps_strings->ps_nargvstr, ps_strings->ps_argvstr, environ));
}
/environ                                                    300,9    8253    87%
```

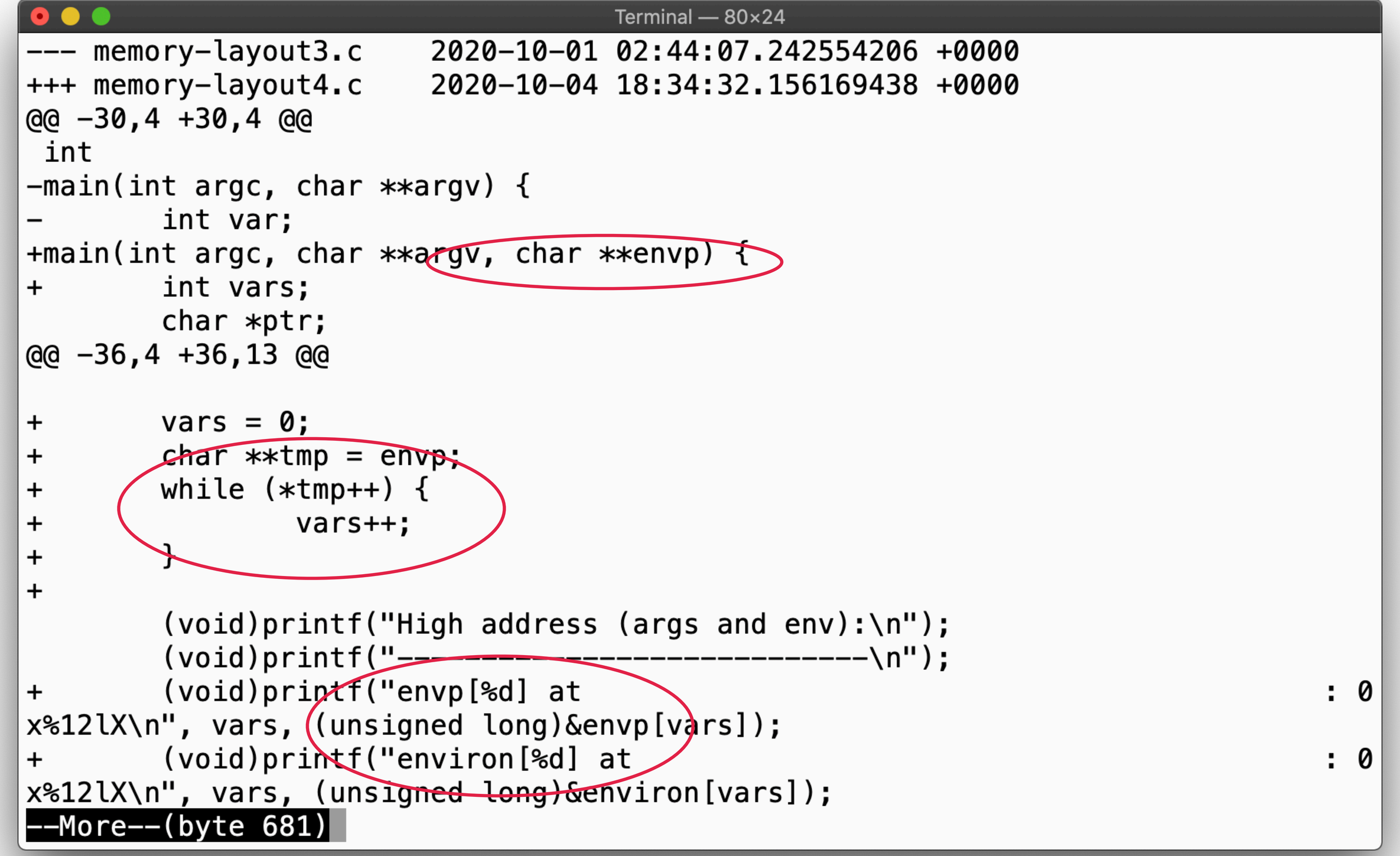
Environment List

Environment variables are stored in a global, `NULL` terminated array of pointers:

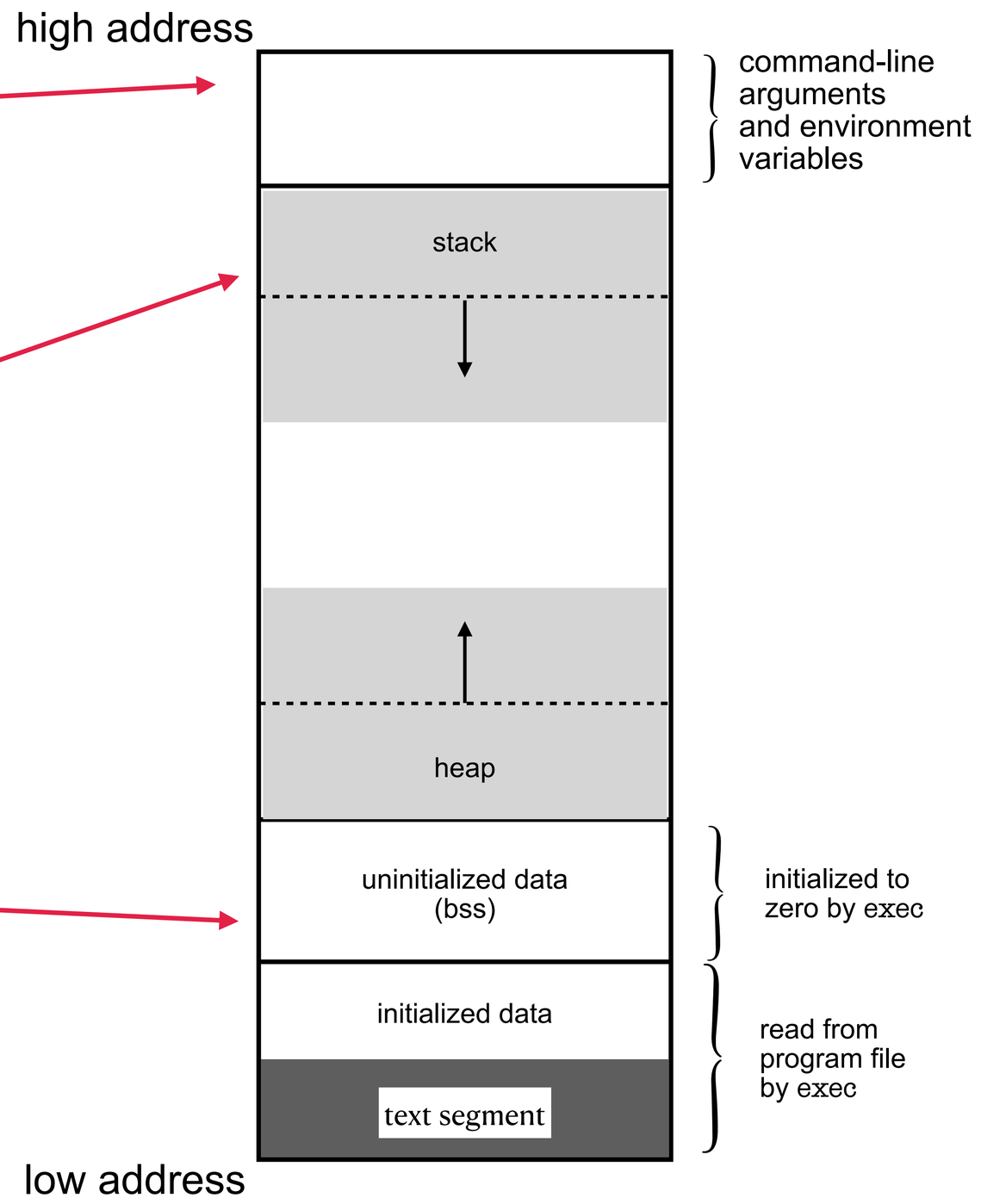
```
extern char **environ;
```

A similar array may also have been passed into your `main` function:

```
int main(int argc, char **argv, char **envp);
```


Terminal — 80×24
--- memory-layout3.c 2020-10-01 02:44:07.242554206 +0000
+++ memory-layout4.c 2020-10-04 18:34:32.156169438 +0000
@@ -30,4 +30,4 @@
int
-main(int argc, char **argv) {
- int var;
+main(int argc, char **argv, char **envp) {
+ int vars;
char *ptr;
@@ -36,4 +36,13 @@
+ vars = 0;
+ char **tmp = envp;
+ while (*tmp++) {
+ vars++;
+ }
+
(void)printf("High address (args and env):\n");
(void)printf("-----------------------------\n");
+ (void)printf("envp[%d] at : 0
x%12lX\n", vars, (unsigned long)&envp[vars]);
+ (void)printf("environ[%d] at : 0
x%12lX\n", vars, (unsigned long)&environ[vars]);
--More--(byte 681)

```
Terminal — 80×49
apue$ ./a.out
High address (args and env):
----------------------------
envp[17] at                                          : 0x7F7FFF66D860
environ[17] at                                       : 0x7F7FFF66D860
envp[0] at                                           : 0x7F7FFF66D7D8
environ[0] at                                        : 0x7F7FFF66D7D8
last arg at                                          : 0x7F7FFF66D7D0
first arg at                                         : 0x7F7FFF66D7C8

Stack:
------
First variable inside main at                        : 0x7F7FFF66D76C
func_array[] ends at                                 : 0x7F7FFF66D760
func_array[] (like 'array[]', but on stack) begins at : 0x7F7FFF66D750
argc at                                              : 0x7F7FFF66D74C
argv at                                              : 0x7F7FFF66D740
envp at                                              : 0x7F7FFF66D738
func2 (from main): frame at                          : 0x7F7FFF66D71C
func frame at                                        : 0x7F7FFF66D714
static int n within func at                          : 0x      601C64
func (called     1 times): frame at                  : 0x7F7FFF66D714
func2 (from func): frame at                          : 0x7F7FFF66D6EC

Heap:
-----
malloced area ends at                                : 0x759D760BF020
malloced area begins at                              : 0x759D760BF000

Uninitialized Data (BSS):
-------------------------
array[] ends at                                      : 0x      601CF0
array[] (uninitialized, fixed-size char * on BSS) from : 0x      601CE0
num2 (uninitialized global int) at                   : 0x      601CD8
string2 (uninitialized global char *) at             : 0x      601CD0
extern **environ at                                  : 0x      601CC0

Initialized Data:
-----------------
num (initialized global int) at                      : 0x      601C60
string (initialized global char *) at                : 0x      601C58

Text Segment:
-------------
func2 (function) at                                  : 0x      400F3F
func (function) at                                   : 0x      400EB8
main (function) at                                   : 0x      400B1A
```



getenv(3) etc.

```
#include <stdlib.h>
char *getenv(const char *name);
```

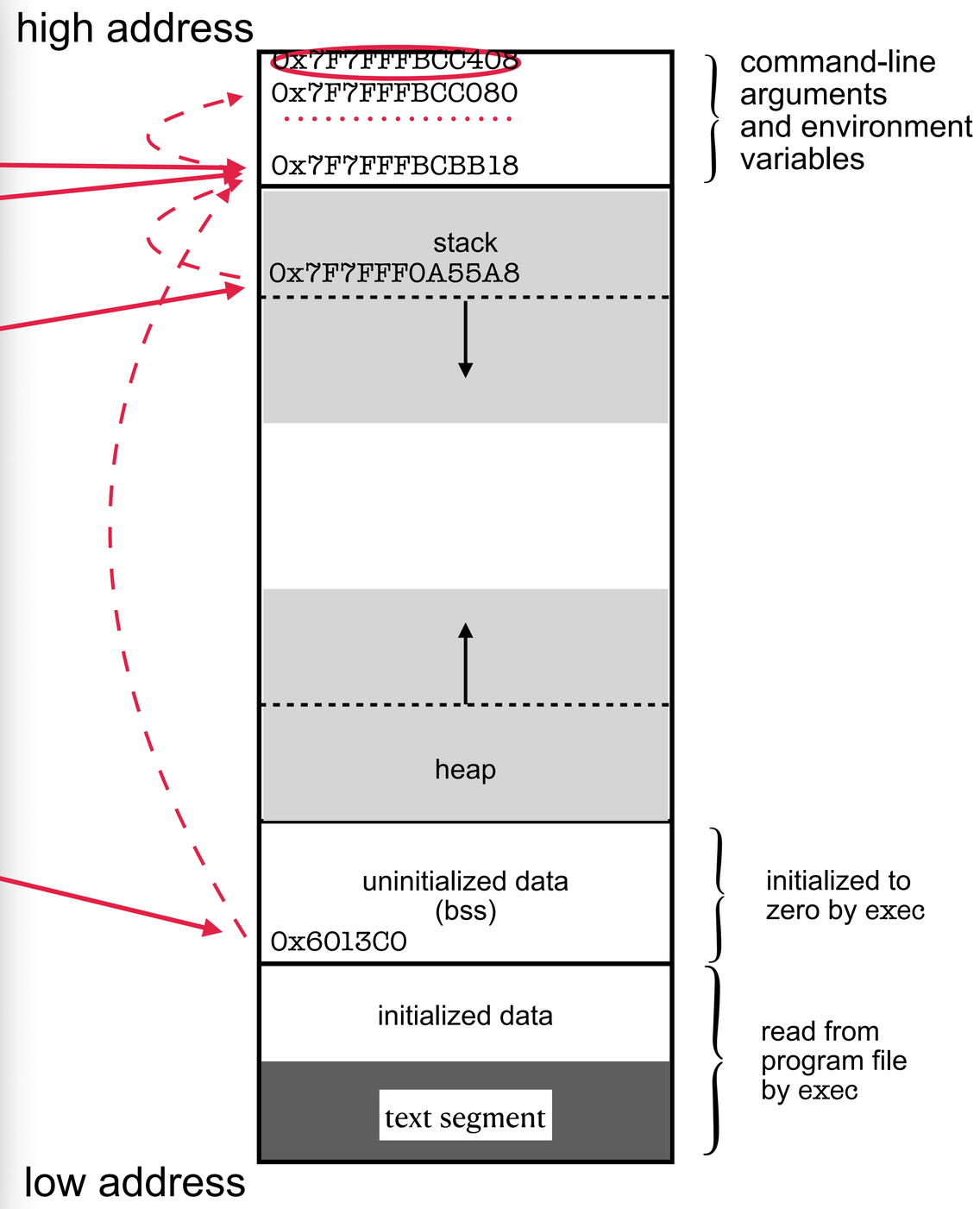
Returns: value if found, NULL otherwise

```
int putenv(char *string);
int setenv(const char *name, const char *value, int overwrite);
int unsetenv(const char *name);
```

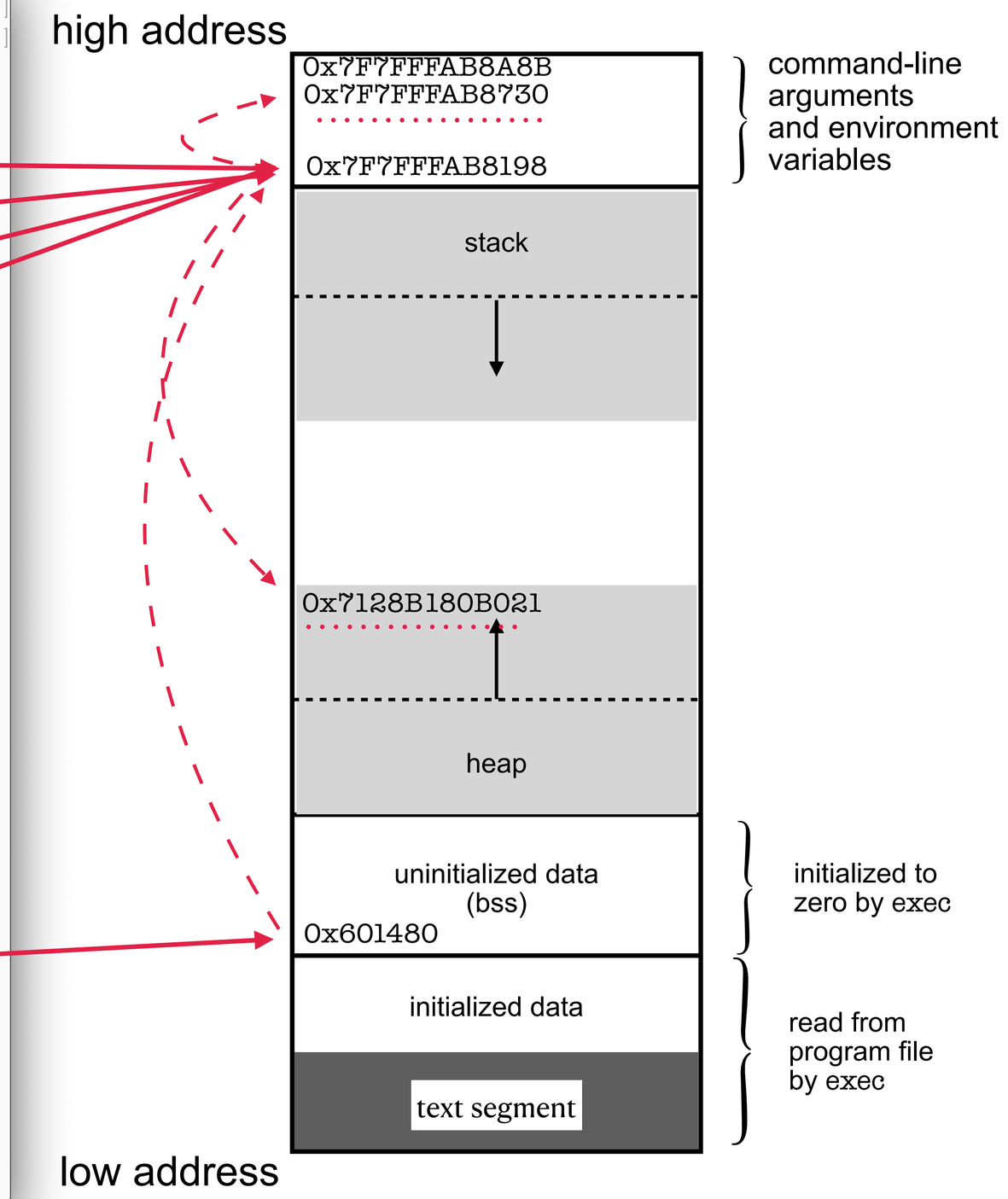
Returns: 0 on success; -1 on error

Note the difference between `unsetenv("VAR")` and `setenv("VAR", "", 1)`!

```
Terminal — 80×44
apue$ cc -Wall -Werror -Wextra memory-layout5.c
apue$ ./a.out
At program start:
0x7F7FFFBCBBA8  envp    [18]
0x7F7FFFBCBBA0  envp    [17]    (0x7F7FFFBCC40B 'SHELL=/bin/sh')
0x7F7FFFBCBB18  envp    [0]     (0x7F7FFFBCC0B0 'FOO=bar')

0x7F7FFFBCBBA8  environ[18]
0x7F7FFFBCBBA0  environ[17]    (0x7F7FFFBCC40B 'SHELL=/bin/sh')
0x7F7FFFBCBB18  environ[0]     (0x7F7FFFBCC0B0 'FOO=bar')

Stack:
------
0x7F7FFFBCBAAC  int argc
0x7F7FFFBCBAA0  char **argv
0x7F7FFFBCBA98  char **envp

Heap:
-----
0x77B13E0BD020  malloced area ends
0x77B13E0BD000  malloced area begins

Uninitialized Data (BSS):
-------------------------
0x      6013C0  extern char **environ

apue$
```



```
Terminal — 80×44
apue$ cc -Wall -Werror -Wextra memory-layout6.c
apue$ ./a.out
At program start:
0x7F7FFFAB8228  envp    [18]
0x7F7FFFAB8220  envp    [17]    (0x7F7FFFAB8A8B 'SHELL=/bin/sh')
0x7F7FFFAB8198  envp    [0]     (0x7F7FFFAB8730 'FOO=bar')

0x7F7FFFAB8228  environ[18]
0x7F7FFFAB8220  environ[17]    (0x7F7FFFAB8A8B 'SHELL=/bin/sh')
0x7F7FFFAB8198  environ[0]     (0x7F7FFFAB8730 'FOO=bar')

After setenv(3):
0x7F7FFFAB8228  envp    [18]
0x7F7FFFAB8220  envp    [17]    (0x7F7FFFAB8A8B 'SHELL=/bin/sh')
0x7F7FFFAB8198  envp    [0]     (0x7128B180B021 'FOO=a longer value')

0x7F7FFFAB8228  environ[18]
0x7F7FFFAB8220  environ[17]    (0x7F7FFFAB8A8B 'SHELL=/bin/sh')
0x7F7FFFAB8198  environ[0]     (0x7128B180B021 'FOO=a longer value')

Stack:
------
0x7F7FFFAB812C  int argc
0x7F7FFFAB8120  char **argv
0x7F7FFFAB8118  char **envp

Heap:
-----
0x7128B180A020  malloced area ends
0x7128B180A000  malloced area begins

Uninitialized Data (BSS):
-------------------------
0x      6014C0  extern char **environ

apue$
```



malloc(3) and friends

```
#include <stdlib.h>
void *malloc(size_t size);
void *calloc(size_t number, size_t size);
void *realloc(void *ptr, size_t size);
```

Returns: pointer on success, NULL otherwise

```
void free(void *ptr);
```

malloc(3) allocates size bytes of uninitialized memory.

calloc(3) allocates number * size bytes of memory, initialized to zero bytes.

realloc(3) lets you grow or shrink a previously allocated area, possibly returning a new pointer.

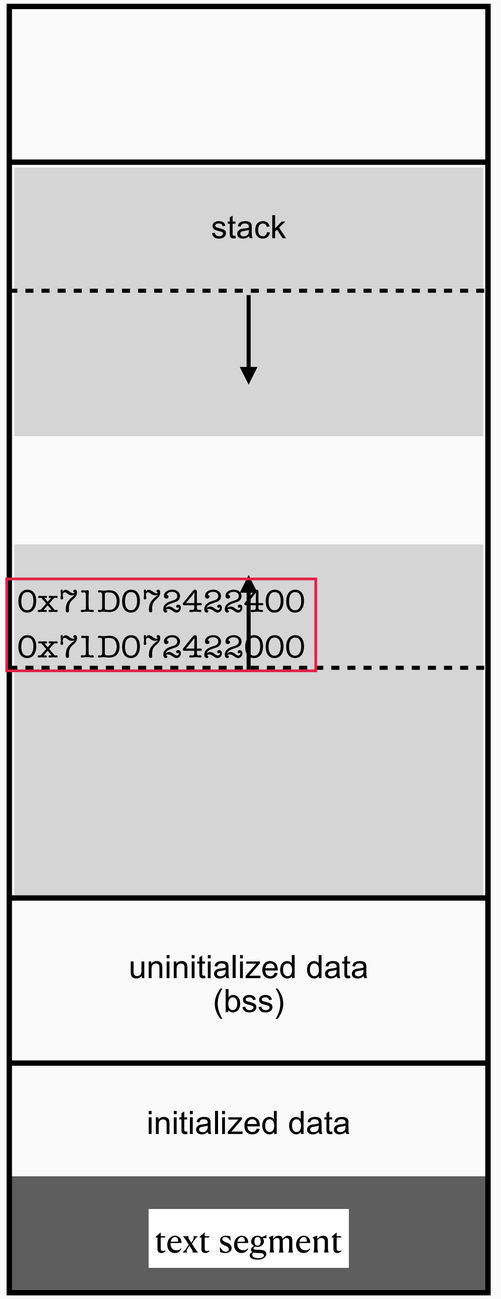
See jemalloc(3) for a lot more details.

Terminal — 65×44

```
#define MALLOC_SIZE 1024 * 1024

void
printMalloc(void *p, int s) {
        (void)printf("malloc/realloc %d\n", s);
        (void)printf("begins at 0x%12lX\n", (unsigned long)p);
        (void)printf("ends   at 0x%12lX\n", (unsigned long)p+s);
        (void)printf("\n");
}

int
main() {
        void *ptr;

        if ((ptr = malloc(BUFSIZ)) == NULL) {
                err(EXIT_FAILURE, "malloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, BUFSIZ);

        /* shrink */
        if ((ptr = realloc(ptr, BUFSIZ / 2)) == NULL) {
                err(EXIT_FAILURE, "realloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, BUFSIZ / 2);

        /* grow */
        if ((ptr = realloc(ptr, MALLOC_SIZE)) == NULL) {
                err(EXIT_FAILURE, "realloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, MALLOC_SIZE);

        /* shrink, but larger than initial allocation */
        if ((ptr = realloc(ptr, BUFSIZ * 2)) == NULL) {
                err(EXIT_FAILURE, "realloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, BUFSIZ * 4);

        return EXIT_SUCCESS;
}
apue$
```

Terminal — 65×44

```
apue$ cc -Wall -Werror -Wextra malloc.c
apue$ ./a.out
malloc/realloc 1024
begins at 0x71D072422000
ends   at 0x71D072422400

malloc/realloc 512
begins at 0x71D072415000
ends   at 0x71D072415200

malloc/realloc 1048576
begins at 0x71D0716FF2C0
ends   at 0x71D0717FF2C0

malloc/realloc 4096
begins at 0x71D072413000
ends   at 0x71D072414000

apue$
```

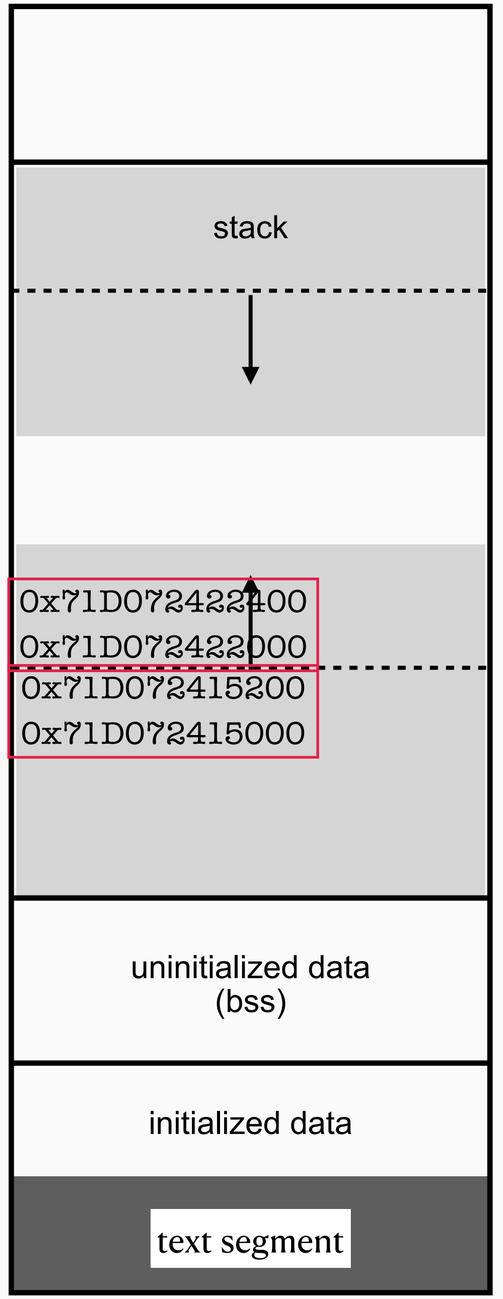


Terminal — 65×44

```
#define MALLOC_SIZE 1024 * 1024

void
printMalloc(void *p, int s) {
        (void)printf("malloc/realloc %d\n", s);
        (void)printf("begins at 0x%12lX\n", (unsigned long)p);
        (void)printf("ends   at 0x%12lX\n", (unsigned long)p+s);
        (void)printf("\n");
}

int
main() {
        void *ptr;

        if ((ptr = malloc(BUFSIZ)) == NULL) {
                err(EXIT_FAILURE, "malloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, BUFSIZ);

        /* shrink */
        if ((ptr = realloc(ptr, BUFSIZ / 2)) == NULL) {
                err(EXIT_FAILURE, "realloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, BUFSIZ / 2);

        /* grow */
        if ((ptr = realloc(ptr, MALLOC_SIZE)) == NULL) {
                err(EXIT_FAILURE, "realloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, MALLOC_SIZE);

        /* shrink, but larger than initial allocation */
        if ((ptr = realloc(ptr, BUFSIZ * 2)) == NULL) {
                err(EXIT_FAILURE, "realloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, BUFSIZ * 4);

        return EXIT_SUCCESS;
}
apue$
```

Terminal — 65×44

```
apue$ cc -Wall -Werror -Wextra malloc.c
apue$ ./a.out
malloc/realloc 1024
begins at 0x71D072422000
ends   at 0x71D072422400

malloc/realloc 512
begins at 0x71D072415000
ends   at 0x71D072415200

malloc/realloc 1048576
begins at 0x71D0716FF2C0
ends   at 0x71D0717FF2C0

malloc/realloc 4096
begins at 0x71D072413000
ends   at 0x71D072414000

apue$
```

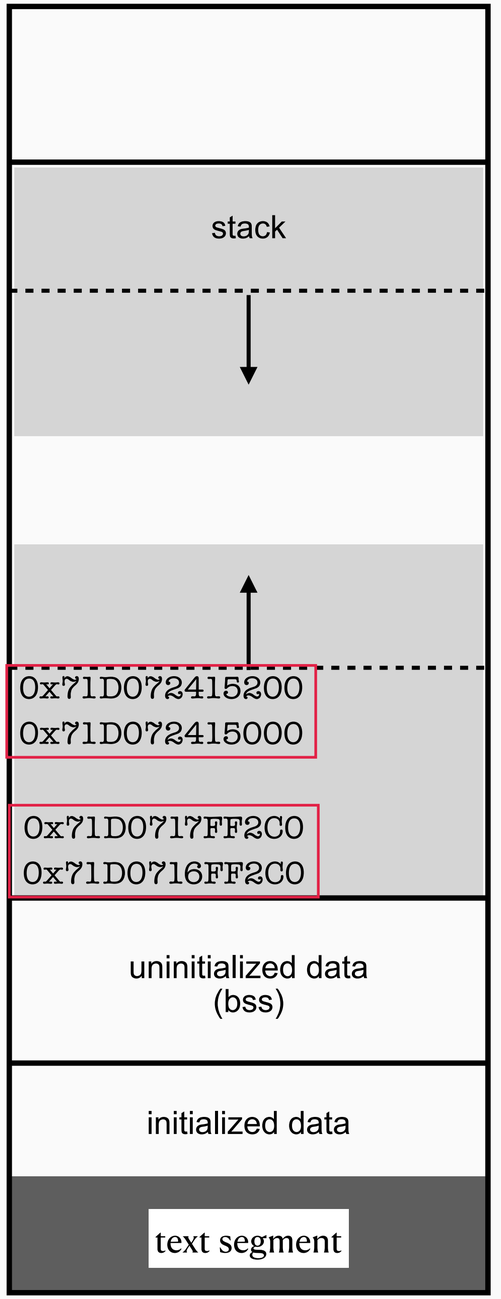


Terminal — 65×44

```
#define MALLOC_SIZE 1024 * 1024

void
printMalloc(void *p, int s) {
        (void)printf("malloc/realloc %d\n", s);
        (void)printf("begins at 0x%12lX\n", (unsigned long)p);
        (void)printf("ends   at 0x%12lX\n", (unsigned long)p+s);
        (void)printf("\n");
}

int
main() {
        void *ptr;

        if ((ptr = malloc(BUFSIZ)) == NULL) {
                err(EXIT_FAILURE, "malloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, BUFSIZ);

        /* shrink */
        if ((ptr = realloc(ptr, BUFSIZ / 2)) == NULL) {
                err(EXIT_FAILURE, "realloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, BUFSIZ / 2);

        /* grow */
        if ((ptr = realloc(ptr, MALLOC_SIZE)) == NULL) {
                err(EXIT_FAILURE, "realloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, MALLOC_SIZE);

        /* shrink, but larger than initial allocation */
        if ((ptr = realloc(ptr, BUFSIZ * 2)) == NULL) {
                err(EXIT_FAILURE, "realloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, BUFSIZ * 4);

        return EXIT_SUCCESS;
}
apue$
```

Terminal — 65×44

```
apue$ cc -Wall -Werror -Wextra malloc.c
apue$ ./a.out
malloc/realloc 1024
begins at 0x71D072422000
ends   at 0x71D072422400

malloc/realloc 512
begins at 0x71D072415000
ends   at 0x71D072415200

malloc/realloc 1048576
begins at 0x71D0716FF2C0
ends   at 0x71D0717FF2C0

malloc/realloc 4096
begins at 0x71D072413000
ends   at 0x71D072414000

apue$
```

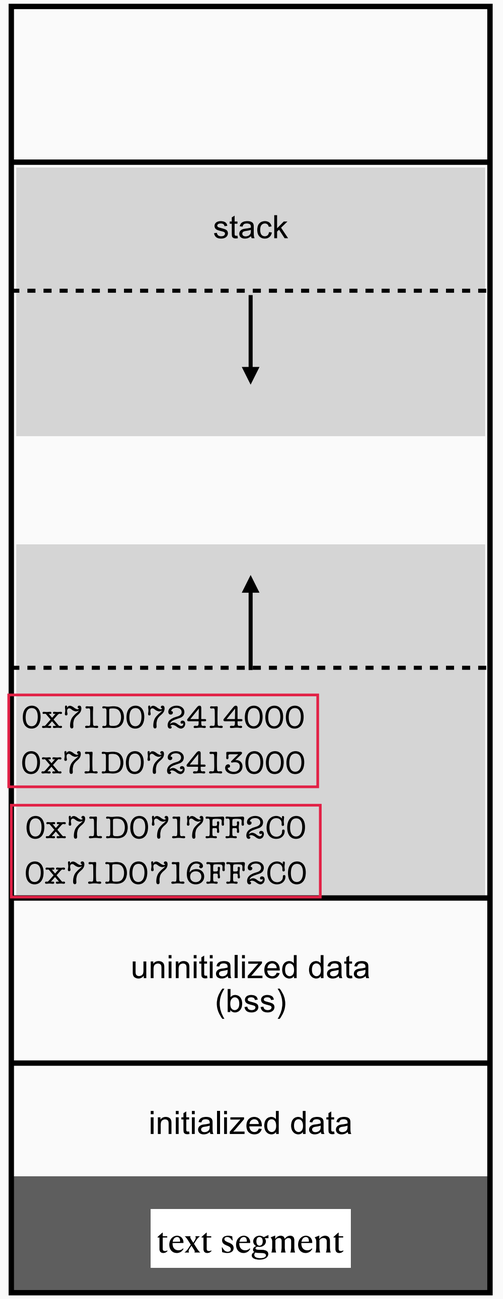


Terminal — 65×44

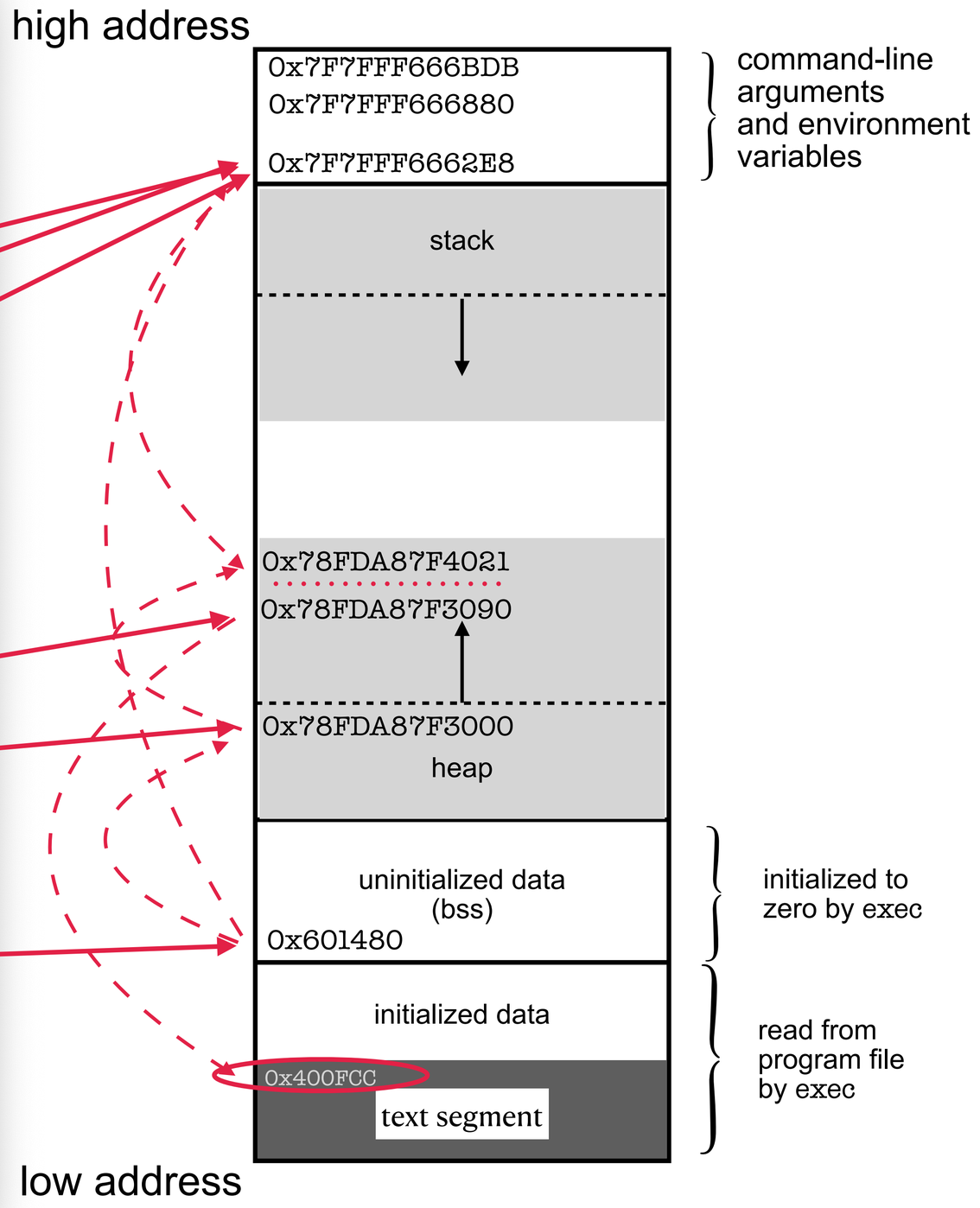
```
#define MALLOC_SIZE 1024 * 1024

void
printMalloc(void *p, int s) {
        (void)printf("malloc/realloc %d\n", s);
        (void)printf("begins at 0x%12lX\n", (unsigned long)p);
        (void)printf("ends   at 0x%12lX\n", (unsigned long)p+s);
        (void)printf("\n");
}

int
main() {
        void *ptr;

        if ((ptr = malloc(BUFSIZ)) == NULL) {
                err(EXIT_FAILURE, "malloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, BUFSIZ);

        /* shrink */
        if ((ptr = realloc(ptr, BUFSIZ / 2)) == NULL) {
                err(EXIT_FAILURE, "realloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, BUFSIZ / 2);

        /* grow */
        if ((ptr = realloc(ptr, MALLOC_SIZE)) == NULL) {
                err(EXIT_FAILURE, "realloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, MALLOC_SIZE);

        /* shrink, but larger than initial allocation */
        if ((ptr = realloc(ptr, BUFSIZ * 2)) == NULL) {
                err(EXIT_FAILURE, "realloc");
                /* NOTREACHED */
        }
        printMalloc(ptr, BUFSIZ * 4);

        return EXIT_SUCCESS;
}
apue$
```

Terminal — 65×44

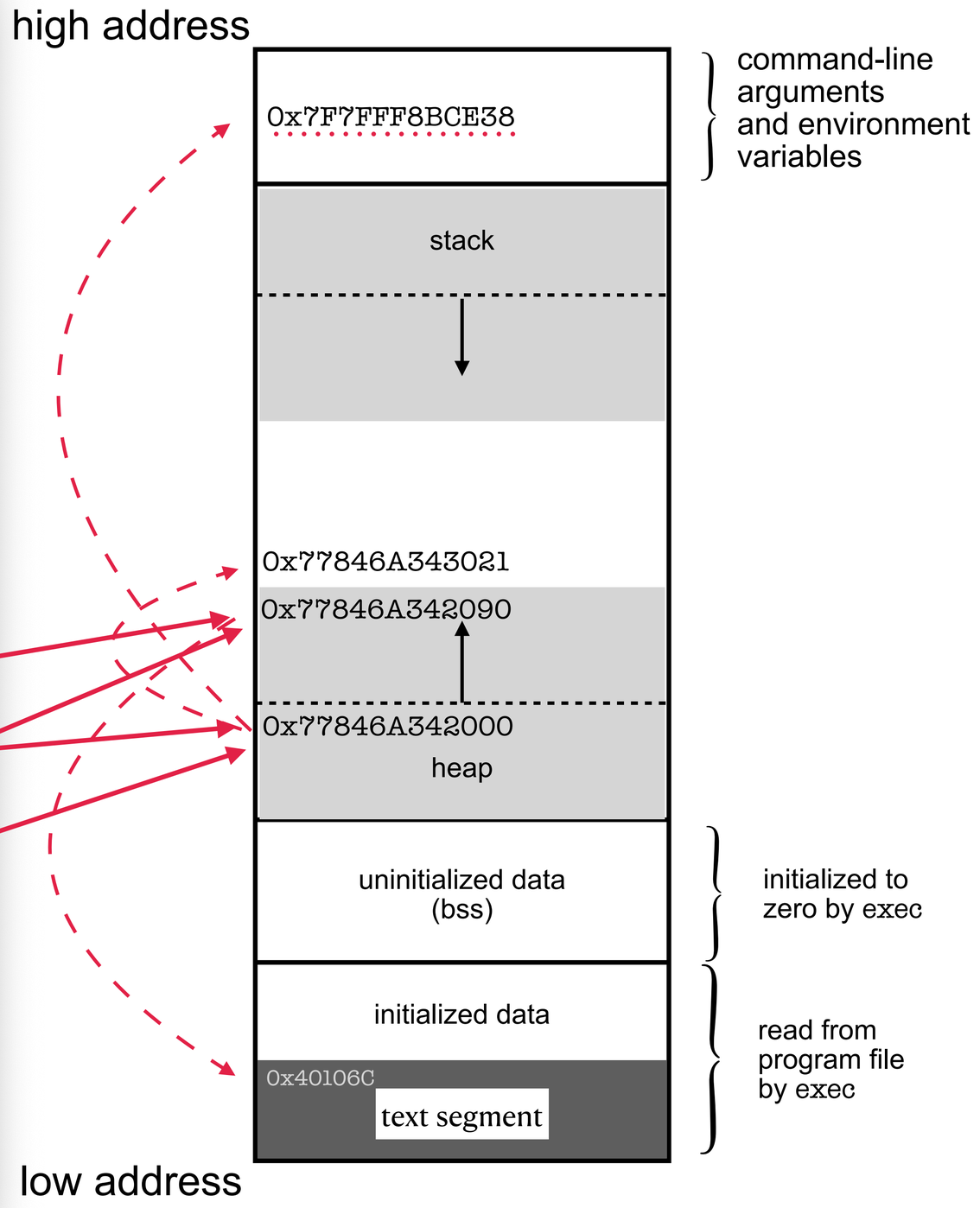
```
apue$ cc -Wall -Werror -Wextra malloc.c
apue$ ./a.out
malloc/realloc 1024
begins at 0x71D072422000
ends   at 0x71D072422400

malloc/realloc 512
begins at 0x71D072415000
ends   at 0x71D072415200

malloc/realloc 1048576
begins at 0x71D0716FF2C0
ends   at 0x71D0717FF2C0

malloc/realloc 4096
begins at 0x71D072413000
ends   at 0x71D072414000

apue$
```



```
Terminal — 80×44
apue$ cc -Wall -Werror -Wextra memory-layout7.c
apue$ ./a.out
At program start:
0x7F7FFF666378  envp    [18]
0x7F7FFF666370  envp    [17]    (0x7F7FFF666BDB 'SHELL=/bin/sh')
0x7F7FFF6662E8  envp    [0]     (0x7F7FFF666880 'FOO=bar')

0x7F7FFF666378  environ[18]
0x7F7FFF666370  environ[17]    (0x7F7FFF666BDB 'SHELL=/bin/sh')
0x7F7FFF6662E8  environ[0]     (0x7F7FFF666880 'FOO=bar')

After setenv(3):
0x7F7FFF666378  envp    [18]
0x7F7FFF666370  envp    [17]    (0x7F7FFF666BDB 'SHELL=/bin/sh')
0x7F7FFF6662E8  envp    [0]     (0x78FDA87F4021 'FOO=a longer value')

0x7F7FFF666378  environ[18]
0x7F7FFF666370  environ[17]    (0x7F7FFF666BDB 'SHELL=/bin/sh')
0x7F7FFF6662E8  environ[0]     (0x78FDA87F4021 'FOO=a longer value')

After putenv(3):
0x7F7FFF666378  envp    [18]
0x7F7FFF666370  envp    [17]    (0x7F7FFF666BDB 'SHELL=/bin/sh')
0x7F7FFF6662E8  envp    [0]     (0x78FDA87F4021 'FOO=a longer value')

0x78FDA87F3098  environ[19]
0x78FDA87F3090  environ[18]    (0x      400FCC 'BAR=another variable')
0x78FDA87F3000  environ[0]     (0x78FDA87F4021 'FOO=a longer value')


Uninitialized Data (BSS):
-------------------------
0x      601480  extern char **environ


Heap:
-----
0x78FDA87F2020  malloced area ends
0x78FDA87F2000  malloced area begins
```



```
Terminal — 80×44
apue$ cc -Wall -Werror -Wextra memory-layout8.c
apue$ ./a.out
At program start:
0x7F7FFF8BC928  envp    [18]
0x7F7FFF8BC920  envp    [17]    (0x7F7FFF8BD18B 'SHELL=/bin/sh')
0x7F7FFF8BC898  envp    [0]     (0x7F7FFF8BCE30 'FOO=bar')

0x7F7FFF8BC928  environ[18]
0x7F7FFF8BC920  environ[17]    (0x7F7FFF8BD18B 'SHELL=/bin/sh')
0x7F7FFF8BC898  environ[0]     (0x7F7FFF8BCE30 'FOO=bar')

After setenv(3):
0x7F7FFF8BC928  envp    [18]
0x7F7FFF8BC920  envp    [17]    (0x7F7FFF8BD18B 'SHELL=/bin/sh')
0x7F7FFF8BC898  envp    [0]     (0x77846A343021 'FOO=a longer value')

0x7F7FFF8BC928  environ[18]
0x7F7FFF8BC920  environ[17]    (0x7F7FFF8BD18B 'SHELL=/bin/sh')
0x7F7FFF8BC898  environ[0]     (0x77846A343021 'FOO=a longer value')

After putenv(3):
0x7F7FFF8BC928  envp    [18]
0x7F7FFF8BC920  envp    [17]    (0x7F7FFF8BD18B 'SHELL=/bin/sh')
0x7F7FFF8BC898  envp    [0]     (0x77846A343021 'FOO=a longer value')

0x77846A342098  environ[19]
0x77846A342090  environ[18]    (0x      40106C 'BAR=another variable')
0x77846A342000  environ[0]     (0x77846A343021 'FOO=a longer value')

After unsetenv(3):
0x7F7FFF8BC928  envp    [18]
0x7F7FFF8BC920  envp    [17]    (0x7F7FFF8BD18B 'SHELL=/bin/sh')
0x7F7FFF8BC898  envp    [0]     (0x77846A343021 'FOO=a longer value')

0x77846A342090  environ[18]
0x77846A342088  environ[17]    (0x      40106C 'BAR=another variable')
0x77846A342000  environ[0]     (0x7F7FFF8BCE38 'ENV=/home/jschauma/.shrc')

Stack:
------
```



The Environment

The process environment consists of an array of strings as `KEY=VAR` pairs.

The initial environment is set up at the top of the process space and pointed to by the `extern char **environ`.

A third argument `char **envp` may be passed to main, which also points to that location.

Setting variables in the environment via `setenv(3)`/`putenv(3)` may lead to shuffling some or all elements of the environ to dynamically allocated memory, but `envp` does *not* get updated!

What programs do with the values in the environment is up to them - but beware, the user can set them to nonsense values!

`https://stevens.netmeister.org/631/env-exercise.html`