Operating System

Blen Daniel Assefa

**OS 2021 Problem Sheet #2**

**Problem 2.1:**  Memory segments

#include <stdlib.h>

#include <string.h>

#include <stdio.h>

char \*strdup(const char \*s)

{

char \*p = NULL;

size\_t len;

if (s)

{

len = strlen(s);

p = malloc(len + 1);

if (p)

{

strcpy(p, s);

}

}

return p;

}

int main()

{

static char m[] = "Hello World!";

char \*p = strdup(m);

if (!p)

return EXIT\_FAILURE;

return (puts(p) == EOF);

}

Text segment:

* The text segment contains the machine code for strdup, main, and all library functions used

Date segment:

* The following variables are stored in the initialized data segments:
  + m(since it’s static)

Heap segment:

* contains the malloced buffer pointed to by p

Stack segment:

* len and s are stored in the stack frame of strdup.

Please refer to the next example for further classification.

static int foo(int a)

{

static int b = 5;

int c;

c = a \* b;

b += b;

return c;

}

int main(int argc, char \*argv[])

{

return foo(foo(1));

}

Text segment:

* The text segment contains the machine code for foo, and main. (If there were library functions used, they would be included here)

Date segment:

* The following variables are stored in the initialized data segments:
  + b(since it’s static)

Heap segment:

* No malloced buffer is used or pointed to, so no heap segment.

Stack segment:

* a and c are stored in the stack frame a foo() function call.
* argc and argv are stored in the stack frame of the main() function call.

**Problem 2.2:**  xargs - execute a programs with constructed argument lists

Inside the folder you will find the code.