Team 4
CIS 657 – Operating Systems
April 21, 2020

Lab 2

Instructions:

Write your own code in the main.c in the system directory in the xinu OS to perform the following:

You need to create two processes with any names such as m1 and m2 or signaler and waiter. One process prints numbers and the other process prints a sentence such as "wow I am running". At the beginning, the first process runs for 21 times to print 21 numbers then it will stop due to wait for semaphore. Then the other process will run to print its sentence. Then you are required to run the first process to print only 5 numbers not 21 anymore, then stop the first process after printing 5 numbers and run the second process to print its sentence once and so on.

Sample of the output:

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 signaler is running 22 23 24 25 26
signaler is running 27 28 29 30 31 signaler is running 32 33 34 35 36 signaler is running . . .
```

Questions:

No questions for this lab.

Modified Code:

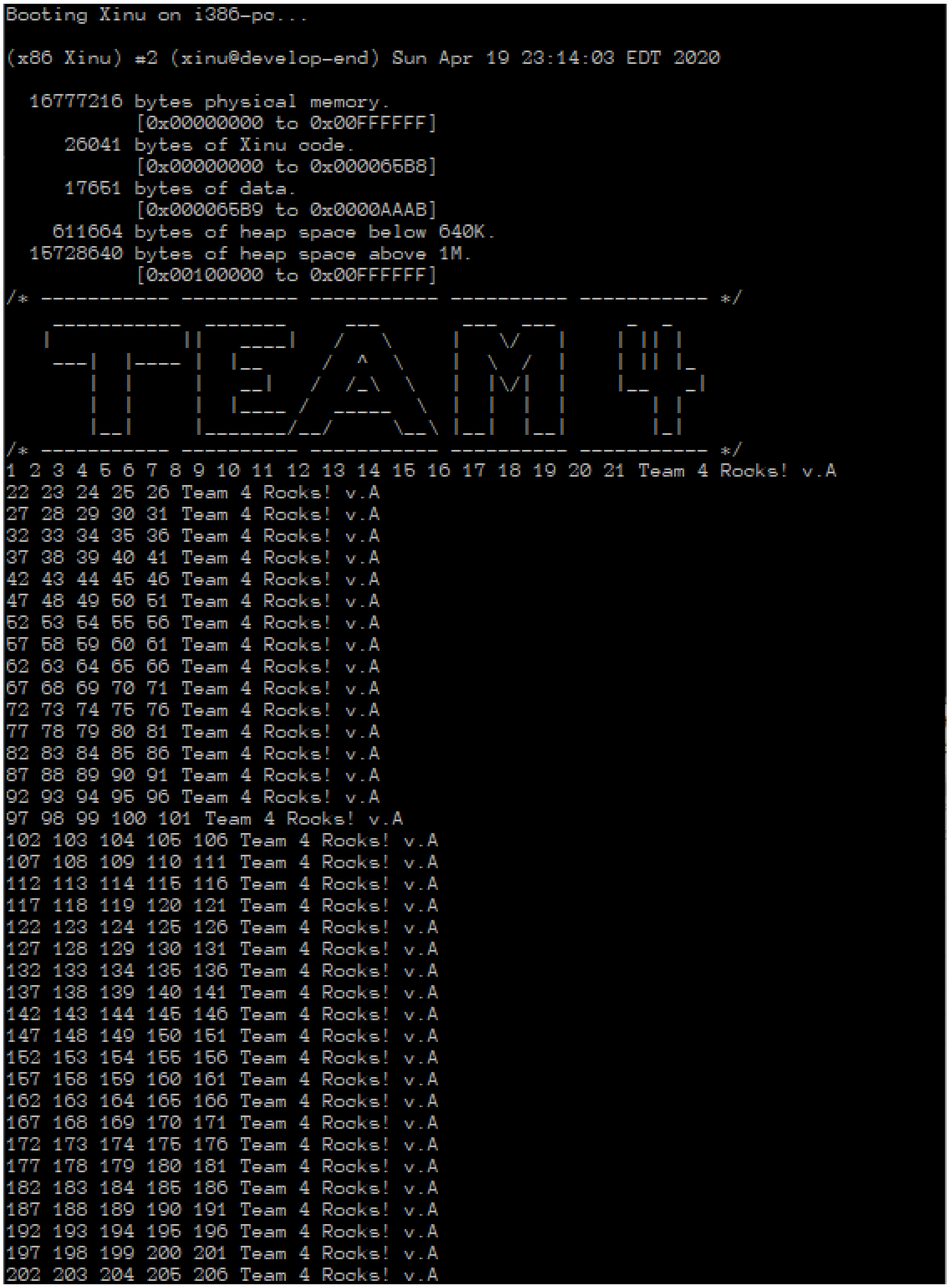
```
/*  main.c  - main */
#include <xinu.h>
#include <stdio.h>

// function/process declaration.
void number_waiter(int32);
void sentance_signaler(int32, int32);
void number_others(int32);
void sentance_others(int32);

// global values to work with.
int32 g_counter = 0;
pid32 g_pid_number;
pid32 g_pid_sentance;
sid32 g_sid_semaphore;

// global define values.
#define _COUNTER_FIRST 21
#define _COUNTER_NEXT 5
#define _COUNTER_MAX 2000

/*------------------------------------------------------------------------
```

Sarah Davis <sdavis26@syr.edu> Patrick S McDougle <psmcdoug@syr.edu> Edgardo Antonio Navarro <eanavarr@syr.edu> Richard Pfautch Jr <rpfautch@syr.edu> Sameer Rizvi <srizvi@syr.edu>

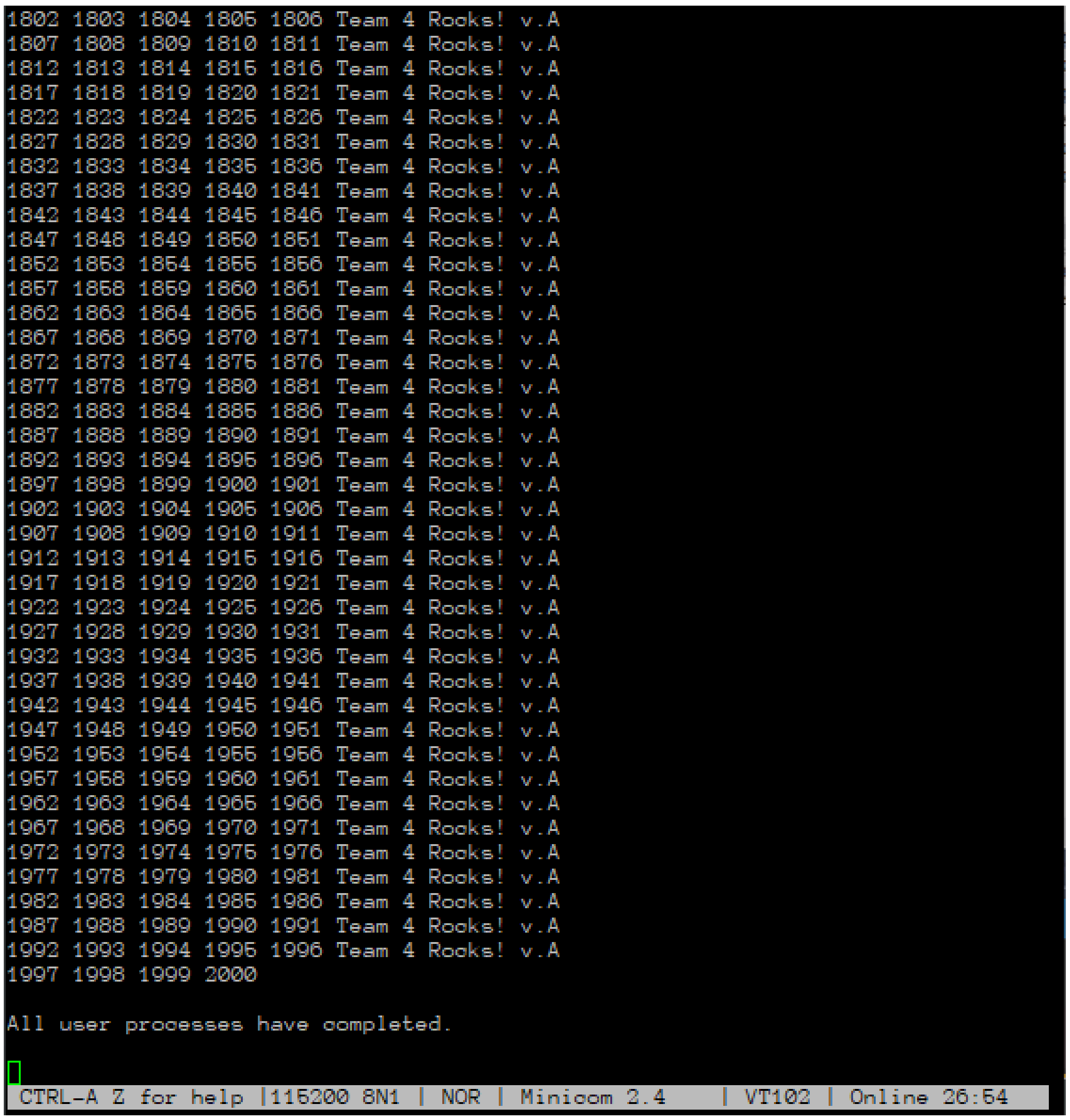
```
 *  main - The one that starts it all.
 *------------------------------------------------------------------------
 */
int main(int argc, char const *argv[])
{
    kprintf("/* ----------- ---------- ----------- ---------- ----------- */\n");
    kprintf("    ___________  _______     ___       ___  ___      _  _    \n");
    kprintf("   |           ||   ____|   /   \\     |   \\/   |    | || |   \n");
    kprintf("    ---|  |---- |  |__     /  ^  \\    |  \\  /  |    | || |_  \n");
    kprintf("       |  |     |   __|   /  /_\\  \\   |  |\\/|  |    |__   _| \n");
    kprintf("       |  |     |  |____ /  _____  \\  |  |  |  |       | |   \n");
    kprintf("       |__|     |_______/__/     \\__\\ |__|  |__|       |_|   \n");
    kprintf("/* ----------- ---------- ----------- ---------- ----------- */\n");

    g_sid_semaphore = semcreate(_COUNTER_FIRST - 1);

    g_pid_number = create(number_waiter, 1024, 11, "print numbers waiter", 1, _COUNTER_MAX);
    g_pid_sentance = create(sentance_signaler, 1024, 10, "print sentance signaler", 2, _COUNTE
R_MAX, _COUNTER_NEXT);

    // g_pid_number = create(number_others, 1024, 11, "print numbers other", 1, _COUNTER_MAX);
    // g_pid_sentance = create(sentance_others, 1024, 10, "print sentance other", 1, _COUNTER_
NEXT);

    resume(g_pid_number);
    resume(g_pid_sentance);

    return OK;
}

/* *********** *********** ********** ********** *********** *********** *
 * Version 1 (A) Patrick's version of the code.
 * *********** *********** ********** ********** *********** *********** */

/*------------------------------------------------------------------------
 *  number_waiter - based of the instructions in the lab.
 *------------------------------------------------------------------------
 */
void number_waiter(
    int32 max_count /* This is the max value that the counter will go to. */
)
{
    for (g_counter = 1; g_counter <= max_count; ++g_counter)
    {
        kprintf("%d ", g_counter);
        wait(g_sid_semaphore);
    }
}

/*------------------------------------------------------------------------
 *  sentance_signaler - based of the instructions in the lab.
 *------------------------------------------------------------------------
 */
void sentance_signaler(
    int32 max_count, /* This is the max value that the counter will go to. */
    int32 num_count  /* This is the number of times the numbers should print after initial set
. */
)
{
    do
    {
        kprintf("Team 4 Rocks! v.A\n");
        signaln(g_sid_semaphore, num_count);
    } while (g_counter <= max_count);
}

/* *********** *********** ********** ********** *********** *********** *
 * Version 4 (D) Other team member's version of the code.
 * *********** *********** ********** ********** *********** *********** */

/*------------------------------------------------------------------------
 *  number_others - based on what others did in the group.
 *------------------------------------------------------------------------
 */
void number_others(
    int32 max_count /* This is the max value that the counter will go to. */
)
{
    for (g_counter = 1; g_counter <= max_count; ++g_counter)
    {
        kprintf("%d ", g_counter);
        wait(g_sid_semaphore);
    }
    kill(g_pid_sentance);
}

/*------------------------------------------------------------------------
 *  sentance_others - based on what others did in the group.
 *------------------------------------------------------------------------
 */
void sentance_others(
    int32 num_count /* This is the number of times the numbers should print after initial set.
 */
)
{
    while (TRUE)
    {
        kprintf("Team 4 Rocks! v.D\n");
        signaln(g_sid_semaphore, num_count);
    }
}
```

Sarah Davis <sdavis26@syr.edu>
Patrick S McDougle <psmcdoug@syr.edu>
Edgardo Antonio Navarro <eanavarr@syr.edu>
Richard Pfautch Jr <rpfautch@syr.edu>
Sameer Rizvi <srizvi@syr.edu>

Output:



```
Booting Xinu on i386-pc...

(x86 Xinu) #2 (xinu@develop-end) Sun Apr 19 23:14:03 EDT 2020

  16777216 bytes physical memory.
           [0x00000000 to 0x00FFFFFF]
     26041 bytes of Xinu code.
           [0x00000000 to 0x000065B8]
     17651 bytes of data.
           [0x000065B9 to 0x0000AAAB]
    611664 bytes of heap space below 640K.
  15728640 bytes of heap space above 1M.
           [0x00100000 to 0x00FFFFFF]
/* ----------- ---------- ----------- ---------- ----------- */
    ___________ _______     ___       ___  ___     _  _
   |          ||   ____|   /   \     |   \/   |   | || |
    ---|  |----|  |__     /  ^  \    |  \  /  |   | || |_
       |  |    |   __|   /  /_\  \   |  |\/|  |   |__   _|
       |  |    |  |____ /  _____  \  |  |  |  |      | |
       |__|    |_______/__/     \__\ |__|  |__|      |_|
/* ----------- ---------- ----------- ---------- ----------- */
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 Team 4 Rocks! v.A
22 23 24 25 26 Team 4 Rocks! v.A
27 28 29 30 31 Team 4 Rocks! v.A
32 33 34 35 36 Team 4 Rocks! v.A
37 38 39 40 41 Team 4 Rocks! v.A
42 43 44 45 46 Team 4 Rocks! v.A
47 48 49 50 51 Team 4 Rocks! v.A
52 53 54 55 56 Team 4 Rocks! v.A
57 58 59 60 61 Team 4 Rocks! v.A
62 63 64 65 66 Team 4 Rocks! v.A
67 68 69 70 71 Team 4 Rocks! v.A
72 73 74 75 76 Team 4 Rocks! v.A
77 78 79 80 81 Team 4 Rocks! v.A
82 83 84 85 86 Team 4 Rocks! v.A
87 88 89 90 91 Team 4 Rocks! v.A
92 93 94 95 96 Team 4 Rocks! v.A
97 98 99 100 101 Team 4 Rocks! v.A
102 103 104 105 106 Team 4 Rocks! v.A
107 108 109 110 111 Team 4 Rocks! v.A
112 113 114 115 116 Team 4 Rocks! v.A
117 118 119 120 121 Team 4 Rocks! v.A
122 123 124 125 126 Team 4 Rocks! v.A
127 128 129 130 131 Team 4 Rocks! v.A
132 133 134 135 136 Team 4 Rocks! v.A
137 138 139 140 141 Team 4 Rocks! v.A
142 143 144 145 146 Team 4 Rocks! v.A
147 148 149 150 151 Team 4 Rocks! v.A
152 153 154 155 156 Team 4 Rocks! v.A
157 158 159 160 161 Team 4 Rocks! v.A
162 163 164 165 166 Team 4 Rocks! v.A
167 168 169 170 171 Team 4 Rocks! v.A
172 173 174 175 176 Team 4 Rocks! v.A
177 178 179 180 181 Team 4 Rocks! v.A
182 183 184 185 186 Team 4 Rocks! v.A
187 188 189 190 191 Team 4 Rocks! v.A
192 193 194 195 196 Team 4 Rocks! v.A
197 198 199 200 201 Team 4 Rocks! v.A
202 203 204 205 206 Team 4 Rocks! v.A
```



```
1802 1803 1804 1805 1806 Team 4 Rocks! v.A
1807 1808 1809 1810 1811 Team 4 Rocks! v.A
1812 1813 1814 1815 1816 Team 4 Rocks! v.A
1817 1818 1819 1820 1821 Team 4 Rocks! v.A
1822 1823 1824 1825 1826 Team 4 Rocks! v.A
1827 1828 1829 1830 1831 Team 4 Rocks! v.A
1832 1833 1834 1835 1836 Team 4 Rocks! v.A
1837 1838 1839 1840 1841 Team 4 Rocks! v.A
1842 1843 1844 1845 1846 Team 4 Rocks! v.A
1847 1848 1849 1850 1851 Team 4 Rocks! v.A
1852 1853 1854 1855 1856 Team 4 Rocks! v.A
1857 1858 1859 1860 1861 Team 4 Rocks! v.A
1862 1863 1864 1865 1866 Team 4 Rocks! v.A
1867 1868 1869 1870 1871 Team 4 Rocks! v.A
1872 1873 1874 1875 1876 Team 4 Rocks! v.A
1877 1878 1879 1880 1881 Team 4 Rocks! v.A
1882 1883 1884 1885 1886 Team 4 Rocks! v.A
1887 1888 1889 1890 1891 Team 4 Rocks! v.A
1892 1893 1894 1895 1896 Team 4 Rocks! v.A
1897 1898 1899 1900 1901 Team 4 Rocks! v.A
1902 1903 1904 1905 1906 Team 4 Rocks! v.A
1907 1908 1909 1910 1911 Team 4 Rocks! v.A
1912 1913 1914 1915 1916 Team 4 Rocks! v.A
1917 1918 1919 1920 1921 Team 4 Rocks! v.A
1922 1923 1924 1925 1926 Team 4 Rocks! v.A
1927 1928 1929 1930 1931 Team 4 Rocks! v.A
1932 1933 1934 1935 1936 Team 4 Rocks! v.A
1937 1938 1939 1940 1941 Team 4 Rocks! v.A
1942 1943 1944 1945 1946 Team 4 Rocks! v.A
1947 1948 1949 1950 1951 Team 4 Rocks! v.A
1952 1953 1954 1955 1956 Team 4 Rocks! v.A
1957 1958 1959 1960 1961 Team 4 Rocks! v.A
1962 1963 1964 1965 1966 Team 4 Rocks! v.A
1967 1968 1969 1970 1971 Team 4 Rocks! v.A
1972 1973 1974 1975 1976 Team 4 Rocks! v.A
1977 1978 1979 1980 1981 Team 4 Rocks! v.A
1982 1983 1984 1985 1986 Team 4 Rocks! v.A
1987 1988 1989 1990 1991 Team 4 Rocks! v.A
1992 1993 1994 1995 1996 Team 4 Rocks! v.A
1997 1998 1999 2000

All user processes have completed.

 CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.4    | VT102 | Online 26:54
```

Sarah Davis
<sdavis26@syr.edu>

Patrick S McDougle
<psmcdoug@syr.edu>

Edgardo Antonio Navarro
<eanavarr@syr.edu>

Richard Pfautch Jr
<rpfautch@syr.edu>

Sameer Rizvi
<srizvi@syr.edu>