

Syscalls and Userspace

ECE598: Advanced Operating Systems – Homework 6

Spring 2016

Due: Friday, 4 March 2016, 5:00pm

This homework involves fast timers and memory allocation.

1. Download the homework code template

- Download the code from:
`http://web.eece.maine.edu/~vweaver/classes/ece598_2016s/ece598_hw6_code.tar.gz`
- Uncompress the code. On Linux or Mac you can just
`tar -xzf ece598_hw6_code.tar.gz`

2. Update the timer interrupt to run at 64Hz (3pt)

- Update `timer.c` so the interrupt happens at 64Hz.
- Update the blink code so it still blinks at 1Hz.
- Update the time syscall so it still returns number of seconds since boot. Update the `ticks_counter` variable at 64Hz, just have the syscall scale properly to seconds before returning.

3. Memory allocation code (2pts)

- Look at the memory allocation code in the `find_free()` function in `memory.c`. What type of allocation algorithm does it implement?
- How would you change the routine to implement next-fit?

4. Memory Allocation (5pts)

Answer these questions in the README file.

64kB memory

0xf
0xe
0xd
0xc
0xb
0xa
0x9
0x8
0x7
0x6
0x5
0x4
0x3
0x2
0x1
0x0

Memory Usage Bitmap

0000 1011 0000 0101

= Used Memory

= Free Memory

Each page of memory is 4kB

- In the above diagram, how much memory is free?

- (b) If you were allocating a 16kB chunk of memory using the first-fit algorithm, where would it go?
- (c) If you were allocating a 16kB chunk of memory using the best-fit algorithm, where would it go?
- (d) In this case, why might the best fit result be better than the first fit one?
- (e) Would it be possible to allocate a 32kB chunk of RAM?
If not, what could be done to make this possible?
Would the proposed action work if the chunks of memory shown were allocated by a C program using `malloc()`?

5. Submit your work

- Run `make submit` in your code directory and it should make a file called `hw6_submit.tar.gz`. E-mail that file to me as well as the document with the answers to the questions.