# CS3502 Project 01-Part III Spring 2023

# OS Project - Memory Manager

# Learning Objectives:

## The purpose of this assignment is to continue the development examples of setting up an Operating System kernel.

## Submissions will be placed here and they will contribute to the total assignment grade of 100.

## This is the kernel build process and constitutes the first module the Process Loader and Scheduler, a process must be loaded into the correct location in memory for the entry point to map correctly to the process address bindings:

## Name the group members and their OS Section here for those who worked on Part III:

## Name: Section:

## Name: Section:

## Name: Section:

## Name: Section:

**This is an individual assignment; you can work with others but please turn in your own assignment each. Also name who was in your group for Project part III here above.**

**Don’t spend more than 30-60 minutes on this, if you have questions contact me with a message asking to meet and we’ll set up a rendezvous (recall: a rendezvous is where both parties are synchronized on a time and protocol) on TEAMS.**

**If you run into errors also contact or send a message to me. This compilation will likely work under any version of linux and compiler but the output may be a little different but should possess the same structure but I need to know if you get errors on your host.**

**For Example: Ubuntu 23 gcc 12 will produce .s files but the linker will give you an error so you cannot generate final binary *hey\_there*.**

**Files needed (from D2L):**

***10\_scheduler.tar.gz***

***“CS3502S01\_Project\_IV\_Scheduler\_Stub.docx” (this file)***

**PROCESS MANAGER**

## This project MUST be completed in the kernel development environment, not an IDE.

## Open up an x-term (or preferred terminal) in your Virtual Machine. If you are running WSL or some other form of linux these commands should still work.

## Starting with 10\_scheduler.tar.gz provided in D2L:

## Retrieve the file from d2l *10\_scheduler.tar.gz* and place it in your project directory.

## In your terminal change directory to your project directory, wherever it is.

## Extract 10\_scheduler.tar.gz with the command *tar -xvf 10\_scheduler.tar.gz*

## Change directory to *10\_scheduler*.

## If you type the command *ls* you should see something like this:

## Run the command *make clean*

## Run the command *chmod 755 s\_compile.sh e\_compile.sh*

## Run the command *./s\_compile.sh*

## If blue double lines obscure this document’s words you may have to right click the blue lines and set to “Ignore” or “Don’t Check…” to see underscores and space properly. ( ./s\_compile.sh)

## Text Description automatically generated

## Using your favorite editor (vi, gedit, etc…) open the file *screen.s*

## There are several functions with declarations like *get\_offset:* **or** *get\_column:*

## What are the names of the *k\_print* functions declared in *screen.s*. How do they relate to the functions found in the file drivers/screen.c?

## Using your editor, open the file *hello.s*

## Find the entry point *start\_*

## Show (copy and paste) the declaration in this file here, include: scope, type and declaration lines.

## Examine the last few dozen lines of the file. What declarations are found there? Is there a declaration starting with *GCC*?

## If so, what is it?, show it here.

## Locate the entry that contains the text 10\_scheduler.

## Show that entry here as well.

## From the directory again, run the command *make clean.*

## Run the command *./e\_compile*

## Run the command *hexdump -C hey\_there | more*

## You will see output like

## 

## Press <SPACEBAR> to move down a page and examine the binary file “hey\_there”. Look for the text on the right hand block of text where is spells “Process”.

## You can also search text in forward direction by typing: */Process*

## When you have found “Process…” show a screenshot here of what is printed by hexdump.

## Do you recognizes any of the other human readable text. What exacly is being stored at this address here?

## **25-50 points**

## **Additional Bonus Question:**

## In examining hello.c it contains a MAJOR flaw (along with my other issues). Can you identify that flaw? Hint, it has to do with the kernel functions themselves and the command inside of *e\_compile*.

## **10 points**

**Submission Guidelines:**

* No handwritten submission is accepted, always submit within this document file. Exception, If you are submitting code, submit as an ASCII formatted text file such as .java, .py, .cpp, .cs etc. or even .txt. NO ZIP FILES.
* You may include your freehand drawing/image and handwritten scans in the submission. However, the writing and images must be clearly legible.  If I notice them as illegible to me prior to the due date I will let you know to provide clearer version.  This will not change the due date.  It is best to present non-handwritten submissions, generally, as is done in the professional setting.
* Show all work/calculations.
* **Please complete your entire work in a single Word Document and Save the file as: yournetid\_CS3502\_Project0X.docx; otherwise, 25% points will be deducted. For instance, I would save my file as: ogarcia5\_CS3502\_Project0X.docx. Then upload your file in D2L.**
* Please observe the submission due date and time. After the due date there is a 50% penalty for the next 24 hours. Any submission after 24 hours of the due date will be graded at 0%.
* If you include a reference or an image taken from other sources, please cite them appropriately. APA is preferred, but cite them so they can be found.
* If you resubmit, please make sure to attach the file again. Your latest submission before the due date will be graded.
* **There is NO extended deadline for this assignment.**