+----------------------------+

| CS 140 |

| SAMPLE PROJECT |

| DESIGN DOCUMENT |

+-----------------------------+

---- GROUP ----

Krisha Rain Martinez [rainmartinez@gmail.com](mailto:rainmartinez@gmail.com)

Jericho Mamaradlo [mamaradlojn@gmail.com](mailto:mamaradlojn@gmail.com)

Rod Labarete [agarchnash08@gmail.com](mailto:agarchnash08@gmail.com)

---- PRELIMINARIES ----

>> If you have any preliminary comments on your submission, notes for

>> the TAs, or extra credit, please give them here.

(This is a sample design document.)

>> Please cite any offline or online sources you consulted while

>> preparing your submission, other than the Pintos documentation,

>> course text, and lecture notes.

None.

System Calls

---- DATA STRUCTURES ----

In thread.h

/\* used to indicate the child’s status, owned by wait-syscall \*/

struct child\_status

{

tid\_t child\_id;

bool is\_exit\_called;

bool has\_been\_waited;

int child\_exit\_status;

struct list\_elem elem\_child\_status;

};

struct thread

{

...

#ifdef USERPROG

...

/\* direct parent thread id \*/

tid\_t parent\_id;

/\* signal to indicate the child’s executable-loading status:

\* - 0: has not been loaded

\* - -1: load failed

\* - 1: load success \*/

int child\_load\_status;

/\* monitor used to wait the child, owned by wait-syscall and the waiting for

\* child to load executable \*/

struct lock lock\_child;

struct condition cond\_child;

/\* list of children, which should be a list of child\_status struct. Owned by

\* wait-syscall

\*/

struct list children;

Guy

/\* file struct represents the executable of the current thread, introduced

\* to deny the running executable and re-enable the write after thread

\* exits.

\*/

struct file \*exec\_file;

#endif

...

}