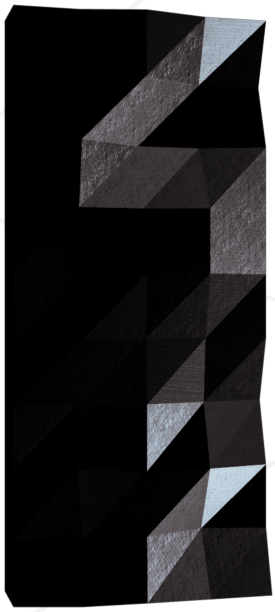

strangely advanced UNIX Project

philosophers

42 sta ff staff@42.fr

Summary: This project is for you to discover threads and mutexes.

Contents

Chapter I

Preamble

Papuan:

In Gentoo, there's daddy Papuans and Papuans not dad ... But some are lice and others without lice .... So in Papuan: there's Papuan Papuan dad there's not dad there's Papuan dad lice there's no dad Papuan Papuan lice there's no dad of lice there's not no daddy Papuan ragweed at lice, lice there's there's dad So not the dad lice among Papuans,

there's Papuan dad dad there's lice Papuan dad dad not
there's not Papuan lice lice dad dad dad there's Papuan not
louse there's no dad Papuan not dad dad lice Papuan
there's no dad dad there's not lice Papuan not no lice dad
dad dad there's not Papuan no lice no dad

If you've got it, well done, is that you know what the binary tree of course, the recursivity ....

chapter II

Philosophers rice eaters

This is the famous story of seven philosophers.

Like all philosophers, they are asking questions. This time, they went ahead not dine with one and only one rod per person. They are convinced that with their clever minds they will share su ffi ment quickly chopsticks to not let one of them die of hunger. The brain of a philosopher consumes eff and lots of sugar. They must eat regularly to avoid falling into hypoglycemia.

If you have not managed to convince them to give up their small challenge, however, you could persuade them to use a simulator of your design rather than risk their health.

Now, their survival depends only on you.

The simulator will meet the following specifications:

- 7 philosophers will be seated around a table.
- Each philosopher has a thread of its own.
- To the left of each philosopher is a wand (so there chopsticks 7 in all).
- A philosophy can not grasp a rod if it is adjacent to it (right or left).
- Two philosophers can not use the same rod at the same time (it seems for hygienic reasons).
- A philosophy involves three steps:
  - Eat with chopsticks 2 (right and left).
  - Rest (do not eat) and holds no wand (even if it is

available).

- Re flexes, in this case the philosopher has one and only one wand in hand.

- After resting state, a philosopher can either eat or re fl ect. If a neighbor already réféchit, the philosopher may acquire common wand, that is to say that a philosopher can have your wand when he re flexes.

- A philosopher who does not eat loses 1 point per SECOND.
- After a philosopher has eaten, he gets all his life and goes to Standby.
- A philosopher whose points drop to 0 dies. At this point the simulation is interrupted until a button is pressed (the choice is yours). This support causes the release of resources and complete stop progarmme.

- Philosophers should not eat in any order if you repeatedly executes the program with the same parameters. They should not eat in a specific order, no scheduler, a philosopher eats as soon as he can. We must respect the randomness of the threads and make do.

- In order to facilitate the defense, you will have defined in your project the following macros:
  - Max_life: The maximum number of hit points of philosophers.
  - EAT_T: The number of SECONDS that puts a philosopher room.
  - REST_T: The number of SECONDS during which a philosopher re pose.
  - THINK_T: The number of SECONDS during which a philosopher re fl e chit.
  - TIMEOUT: The time in SECONDS after which the simulation is interrupted by a ffi singing "Now, it is time ... To DAAAAAAAANCE! ! "If no philosopher has died of hunger (the current dinner must remain visible on screen to the balance sheet). Pressing a button causes the release of resources and a complete cessation of the program.

- You must make a GUI (wx, SDL, ncurses, OpenGL, ...) illustrating clearly what is happening in the program.

Chapter III Practical information

- This project is to achieve in tandem ie 2 and no more.
- This project will be corrected only by humans. So you are free to organize and name your fi les as you wish, nevertheless within the constraints listed here.

- **The executable will be named philo**
- You have to code in C on a standard dump and make a fi Make it with the usual rules.

- **If you are smart and use your driver DLL libft, you must copy the sources and Makefile associated in a folder named libft which should be at the root of your deposit returned. Your Makefile will compile the bookstore, calling his Makefile then compile your project.**

- According to the image library you use, it can be included in your rendering, and as libft it must be compiled automatically by the Make fi principal.
- Your project should be the standard. This is the norminette is authentic.
- You have to handle errors in a reasoned way. In any case your program should exit unexpectedly (Segmentation fault, etc ...).
- **You must go to the root of your deposit made, a file author containing your logins followed a '\ n' each:**

```
$> Cat -e author xlogin
ylogin $ $ $>
```

- As part of your required part, you have the right to use the following functions:
  - malloc.
  - free.
  - write, read.

    - usleep.
    - time.
    - exit.
    - pthread_create.
    - pthread_detach.
    - pthread_join.
    - pthread_mutex_init.
    - pthread_mutex_destroy.
    - pthread_mutex_trylock.
    - pthread_mutex_lock.
    - pthread_mutex_unlock.
    - the functions of the graphic library you selected
    - You have permission to use other functions in your bare cardre bo-, provided that their use is justi fi ed in your defense. There are of course points for the bonuses.

- You are entitled to a maximum of 3 total. Global must be JUSTIFIED.
- You can ask questions on the forum ...
- Good luck to all !