Path : (Tuning)

Algorithm of things which are :
1. Simple. (First Level related)
2. Fundamental.
3. Library
4. Function
5. GNU Tool Chain
6. Operating Systems.
7. Computer Networks.
8. Data
9. Visualization

< core area > : interfacing all the 9 layers.

write a classical book on tuning.
write a book about the art of fixing things in computing.

AN ALGORITHM BECOMES A FUNCTION WHEN USED IN AN APPLICATION.

read about database concepts and typical related things
try to contribute to libmysql library

try to know about <function> of GNU C library. example : sqrt() function

keep one book on geometry
keep one book on applied physics
keep one book on chemistry
keep one book on mathematics

tuning of/with algorithms for an optimal configuration
try to use Gcov.
(try to use a command named "script" for the above
focus on scripts for kernel related. ex: linux-3.10.0/scripts (scripting with Bash)

CONTRIBUTIONS TO THE WORLD WILL BE AS :

sharing scientific knowledge through BW Project

try to maintain a print version of beautifulwork
try to document on beautifulwork about algorithm :
1. Analysis of GNU C library.
2. Testing of GNU C library.
3. Improvement to GNU C Library.

a typical beautifulwork page from top down :

1. learning model (any one of the learning models i have documented)
2. comment(analysis) from algorithms(science) about function/API discussed.
3. Gcov test report
4. comment about the improvement of the algorithm.

try to write code for geometric functions.
try to form a library using those functions.
try to write an application using that library.

if there is a lot of code reading to be done,
i will do it only if i feel that people may use it.
i like to port some GNU software that work in commandline
interface to a web interface (as a web application).

programming blog:

1. you feel tired of programming because you are
trying to program in a framework. you usually have
problem working in a particular time frameworks.
i usually like to play with things than work.
2. try to solve same problem in different programming
languages.
3. try to solve every function in different programming languages.
4. you may get new ideas maybe when you excercise or play with a function.
play means like coding a user defined functions in different languages.
5. try to rewrite the same program again and again so may be
your brain may get comfortable with the patterns.
6.be slow and easy.

notes :

1. read code and write useful scripts for functions.
you can take a function in trueprint program and write
a relevent script for that function. a script that supports
the function in code in a way .Also create a directory called
"scripts" inside parent directory.

2 . you find it easy to work on already created things.
like hacking on already existing programs.

3. my creativity lies in music a lot.

4. my typical scientific thing lies in problem solving(fixing)
skills and maintanence development(writing scripts and system
administration).

5. i like to be a visual artist.
i like to focus on drawing cartoons for beautifulwork.org posts
using GIMP.

I like to be a "DEBUG METHOD DEVELOPER"

get source code of quicksort in different languages.(C,JavaScript,Perl,Python,C++,Java...)
try to debug all the code with debuggers.gdb , may be firebug .. chrome...related...
may be try to integrate or otherwise if possible run the code into different
environments and conditions and debug to see if there is any bug.fix the bug
or give the idea to fix it to the developer if possible.

also try to learn about linux kernel debugging.

take source code from lots of free software(may be debian) and do
extensive debugging.lot of debugging is studied through testing
(i mean atleast testing the debugger features using a program)
note : Testing programs using debugger and debug techniques
make it a test driven development. a development of debug methods.
(DEBUG METHODS DEVELOPER)

collect pdfs and documents about debugging in other format and
try to compile whatever necessary and make a series of books like stuff.
prepare a document with the documents you have used for debugging stuff.

try to document and develop methods for unlocking knowledge.
engineering methods and algorithms for unlocking knowledge.
METHODS Developer.
METHODS Algorithm Developer.
Algorithms For Methods Usage.
Package Developer.

Fix/Debug Method Developer.
types of fixes.
a)Code Styling Fixes
b)Style Fixes. (CSS)
c)Build/Compile Fixes.

be patient and try to fix small and easy fixes
1. unable to change gnome background.

A PROCESS THAT CREATES NECESSITY MAY DRIVE INVENTION.

try reading and learning scripts from free software projects
and get ideas and port the usable parts to GNU (dap+trueprint+gleem)

try creating user interface to create posts in wordpress related to the learning models in beautifulwork.

asking help from people to contribute to beautifulwork.

note down the things you have done in typical free software related
so that you can repeat or hack in a way.

1. source code build.
2. debug using gdb to know internals.
3. General debugging.
4. show commandline session and title it as knowledge representation.
5. file bugreport to free software projects.
6. answer questions on free software mailing lists.
5. install free software operating systems.
6. advocate for free software.
7. create debian(.deb) and also GNU packages.
8. system/network/security administration
9. drawing images using gimp.
10. trying to create icons using inkscape.
11. shell programming.
12. web application programming.
13. learning editors like emacs and vim.
14. computer assembling and maintainence.
15. learning from reading mail from mailing lists.
16. contributing to Linux kernel.

BW Project content cycle:

1. GNU/Linux commands screenshot. (knowledge representation)
2. Learning Model V.
3. Using gdb to find internals of a GNU/Linux command.
4. Embed videos of GNU/Linux command tutorial and related knowledge representation
5. Hacking Web Applications related to Perception Model.
6. Graphic Images about a GNU/Linux command.
7. debug method development. (try to fix bugs from debian BTSand GNU BTS related)
8. Learn, download code, debug , test about algorithms given in the book Encycclopedia of Algorithms.
9. Linux kernel Testing using "autotest software". and do knowledge representation.

in www.beautifulwork.org may be it should categorize whatever
that has been done in it to it's proper scientific subject.
for example command "ls" to operating systems ...

< forming and realizing the team >
may be write down the softwares and people that are related
to www.beautifulwork.org and may be watch them, talk to
them note down their websites and mailing lists and may be also
try to hack the code of the software that beautifulwork.org depends.

search even simple things in google so that you can get the result fast

create posts in beautifulwork.org to explain terminologes like
wrapper and hook

```
TEAM. < it is good to get from and also contribute to the team >
```

Project Name : BeautifulWork Project.
Purpose < Debian Research >
Debug Method Development (Theory Development)
Learning Process.

try to work with the API of all team members like google and youtube related)
the team members listed below may be listed as different projects related

1. Mailing Lists.
2. Google Search
3. Wordpress Project.
4. Debian Project.
5. Linux Kernel.
6. GNU Project.
7. Wordpress Plugins.
8. Youtube.

beautifulwork.org color combinations -- green + white + grey

play + learning

following may not be accurate :
learning model 1 = put the output of "whatis < command >" in post title.
connect beautifulwork.org contents words with existing GNU, Linux and other relevent doc.
try to find reasons why the developed learning models are good (assuming it is good)
try to tell the skills that are developed when working with a typical learning model.

trueangle.org :
try to connect to other wordpress databases other than beautifulwork.org related.
try to connect to wordpress blogs that are about free software and also may be ad free blogs related

people involved related:
inorder for getting people involved...
try to document what ever that has been done in beautifulwork properly
try to document how to contribute to beautifulwork in a detailed step
by step manner.try to document what are the things that needs to be
fixed.

learning model V:
try to publish posts in this model.
post title can be the proc parameter or api parameter name related.
try to include a commandline session of the command that is related
to the proc parameter and the commandline session of the command that
is related to a typical API.

may be change "UNIX" in the learning models to "GNU/Linux or UNIX"
a command may be a function of GNU libc or normal bash commands.
try to work with the book "THE LINUX PROGRAMMING INTERFACE"

new learning model . LM X
1. commandline session.
2. Video Tutorial (also give the link of youtube channel from which the video has been taken , if it is so)

IT MAY BE GOOD TO KNOW THE AREA OF MY LIMITS AND THE AREA WHERE IAM UNLIMITED

different angles of beautifulwork:
(document and organize the entire hack(playful cleverness) session for knowledge building)
1. education with GNU (play and learn) - freedom to use the program for any purpose.
2. free software research to know "how things work" - freedom to study the source code to know how things work.

team culture related:
anybody can do any work.
a person can work in multiple roles among different projects at the same time.
maintainers approval is required for the official release of a patch.

a patch
1. content development patch.
2. software development patch.

Role:
may be initially a person has to take the role of a subscriber.
a person becomes a contributor after his contribution is accepted mainstream.

Different sub projects under beautifulwork project:
1. learning model 1-10 (considered as 10 different projects)
2. document of debug sessions (for debug method development)
3. search development
4. development of data visualization and statistics.
every learning model project will have a administrator.
every learning model can have a team of contributors.

core work :
the core work of beautifulwork is debug method development.
the final outcome of the beautifulwork project is a book on DM
DMD = Debug Methods.

do not do:
do not try to develop any learning algorithms.

learning model in this document means learning styles.

try to build knowledge,experience and methods for debugging.

repeat the same function or command with different input
(this will help in building experience)

function:

learn about "functions" in mathematics (video tutorial may be easy) and try to make meaningful relations between mathematic function and GNU commands.also try to make GNU command or function or algorithms with the help of laws or theorems in the area of "functions" in mathematics by "mapping" between the meaningful relations.

REASON FOR WRITING ANOTHER VERSION OF A PROGRAM CAN BE FOR THE BEAUTY OF DETAILING

try to stay with wordpress in BeautifulWork Project: Be collaborative

try to stay in partnership with other projects

in debug method development try to connect topic of "functions" in mathematics to GNU/Linux commands

another learning model is to display a single image that explains something

document design principle related:

for example " what kind of font should be used where and why ? "
try to make things uniform throughout the site and give detailed documentation of what of what is used in the site.

learning model 10:

1. EXPLANATORY IMAGE
2. SOURCE RELATED (SOURCE OF IMAGE)
3. RELATED FROM RESEARCH PAPER (LINK)

in connecting ("stitched to") option there is a parent-child tree like structure based on necessity suppose topic A have a content which is necessary for topic B . A becomes a parent of B

may be modify the wordpress database design and may it
suitable for your programming

try to contribute to beautifulwork by solving problems in maths for class (4-12)

try to do testing math functions for perl and python and javascript

try to hack core wordpress with the help of code available in the web

try to get patches from mainstream wordpress and apply to beautifulwork version of wordpress related

may be try to work with fonts and kerning related free software projects

if you want to go more core in a way... may be go core in wordpress internals

may be giving supportive link for a post is useful . it may form a collection

may be try to write the balancing act related algorithm and may be try to connect it with every plugin. may be a whole lot of code should emerge from the balancing algorithm

you may be a style user,developer and administrator

change the learning style contents from UNIX command to GNU Command
make the headings in a all learning styles to normal size(not h1 h2 ..)
stay with beautifulwork project for atleast 10 years

it not about doing coding work in the project .
make the project with meaningful content and may be show things
and commands which has a relation to operating system and computer networks

may be make programs and design weaker than the existing one
for learning purpose

may be make the algorithms tuned for a particular problem

may it is possible to write 1000 sorting algorithms tuned for 1000 different problems

doing manual things in software may help in getting idea to automate it

bugs:

1. BUG can occur during software compilation.
2. BUG can occur during porting software.
3. BUG can occur during software installation.
4. BUG can occur during software configuration.

marketing related:

1. try to get reviews online .may be paid reviews.
2. try to get suggestions online.
3. try to contact people and ask about what they like to see in the website.
4. try to contact people and see the features that they like to get added.
5. try to contact different type of people. people from different industry for suggestions.
6. try to contact the online GNU, Linux and other communities for suggestions.

jobs related:

considering suggestion from people and from your own ideas
try to allot jobs to other people for a small fee.

bw project clarity:

try to learn software testing using python.
after that write tests for libeautifulwork.
try to write infinite algorithms for libeautifulwork
try to test for time issue of a program using gprof.
try to do python unit testing for correctness of programs.
try to do profiling using gprof.(may be GNU C programs)
try to port algorithms to linux kernel
try to port algorithms to userspace programs

normal travel is in the breadth and sometimes you go depth but
if you stay in depth for a long time your pressure may increase,
so come back to breadth travel again(do come back well before you explode).

learn linux kernel debugging:

if you find a library that is doing an advanced job in an area than the one your
are creating then try to use that library(use the API if possible) in the development of your library

YOU SHOULD PUT ONE SINGLE IMPERFECTION IN EVERY THING YOU LIKE TO PERFECT

you keep writing algorithms, and your mind will guide you to find out
which algorithm is faster and how to fix algorithms to improve speed.
use "time" command normally.

always try to master fundamentals:

```
Once you master the fundamentals of one topic you can switch on to another topic
may be connecting fundamentals of multiple topics can lead to research
```

topics you may work on: (may be you can use these topics as categories for beautifulwork.org posts)

1. Debian.
2. GNU.
3. Linux Kernel.
4. Operating System.
5. Computer Network.
6. Algorithm and Data Structure.
7. Debugging.

may be you can switch between topics in a random manner

the method of an algorithm which is written by someone else
for the same problem may not be as you expect it while reading their code

i work for free software(and education), people can use me as a resource for their work

when you learn algorithms may be you can try to learn where
those algorithms are used in real life.

you can contribute to a project by learning about the project and make
the necessary notes if possible and make a design (way to) solve the problem

learning by examples is a method used in beautifulwork project

people may contribute to beautifulwork project by contributing examples to the project

topics involved related to basic scientific research may be :
1. Number Theory
2. Human Learning and Education <www.beautifulwork.org>
3. Algorithm - O notation.

basic research may be defined as that undertaken primarily to acquire new knowledge of the underlying foundations of phenomena without regard for a particular application.

benefits of basic research:
1. new knowledge - it is the primary product of basic research.
2. social benefits - the social benefits of educating high quality scientists, technologists and engineers.
3. basic discovery may lead to applications

we may not know what we are going to discover

it may not be necessary to have a complete understanding of a thing to make it work.

when you work on something and if you find it small and not worth working, keep working and try to search and find out the importance of that thing

may be try to bring the output of your work to be easily accessible and understood

if you rework and research on basic things, you may be doing a research on a topic you may not know and also an outcome you are not sure about, but the outcome may be connecting all the sections you work on

when you work on real life problems section of bw project and if you feel that you have not completed a program and you feel that you are drawn towards that program and do not know when to stop typically, then you should try to complete the design of the program using coding and documentation which will be like writing the initial version of the software.

may be add math and physics as sub groups for your basic research.
always concentrate on basic and simple stuff

however simple an algorithm looks, it may better to say "i know it" only if you really implement it in code and make it work.

```
--- may be add a link called "papers" to beautifulwork.org for adding papers you
make using
    libreoffice-writer ---

--- may be try to learn algorithm and data structure using javascript ---

--- may be try to do the documentation for BW.org and add software to contribute
to beautifulwork ---

--- may be fill the contents for the links "copy" and "contribute"  ---

--- may be rework on the "about" page ---
--- application software ---
{


1.try to remove password hashing from public database.
2. program to make user view posts at their font.
3.program to concatenate posts to form a big document.
3.1 option to add posts in between.
4. spell check program for posts
5. option to sort posts on different basis.
6. options to add and remove posts from users private account.
7. content rework.
8. program to search within posts
9. if upstream author is there then details about that post should be there on
every post.

}


--- may be in future try to do API development for your library project  ---

--- may be try to write home management software as a wordpress plugin ---

--- may be consider wordpress as a kernel a plugins are applications ---

--- may be try to do one original algorithm in your library project ---

--- project euler and other mathematics  ---

--- in the HOWTO link of beautifulwork.org you may give instructions to download
    database and learn to set it up locally  in debian and may be other
GNU/Linux
    systems  and use it locally , may be scripts should be developed to download
    and upload content and also update database if possible ---


--- may be try to write a wordpress plugin on how to learn. for exmaple:
    How to learn new things [ start learning in breath and then go to depth ]
    start learning simple things [ You can identify these things if you
categorize simple things  ]
    When to relearn things [ You can relearn things next year in the same day
and also contribute ]
    define levels in learning ---

--- focus on style fixing  and  debug method development ---

--- may be focus on artistic things like ....

    1. Fixing Fonts.
    2. Fixing Animation. JavaScript
    3. Fixing Color. GIMP
    4. Fixing Graphics. GIMP
    5. Fixing Images. GIMP
    6.
                  ---



--- now it is 5 years of beautifulwork project.
    now may be the learning models can be modified ...
    may be concentrate on using  and collab with API's (google API, Wordpress
    API and Youtube API )  for the next 6 months (may be part of openlearn
plugin)
    or may be do algorithm development work for next 6 months ---
```

--- may be afterwards if possible write GIMP Plugins with C Python Scheme ---

--- may be for media lab , topics are CSS, GIMP, Blender, Scratch ---

--- Learning "Processing" and "GIMP" can be treated as basic research on basics ---

--- B openlearn .

try to write code for Book Suggestions.
may be try to write code using GitHub API.
try to write code to accept feedback by the user about learning.
try to write code to give suggestions based on feedback.
your database is your knowledge base related.
may be try to connect the posts in the base and find common things.
may be write code to counsel the learner.

--- may be try to do specialization in coursera related in "Fundamentals of Computing"

reasons : 1. certificate of proof and reputation 2. collaboration with coursera people
3. learning more python 4. learn fundamentals 5. gaining more programming experience
6. capstone project. ---

--- Commandments

0. Learn and Hack free software (Work With BW and Trueangle Projects)

1. Take one feature from different projects and try to add it to BeautifulWork Project
(Do not entirely follow a Project)

2. Find common thing in all your work and try to connect them.
(Knowledge and Learning are common things in BeautifulWork Project)

3. Try to build new knowledge by connecting existing knowledge base.

4. Try to write algorithm to connect existing knowledge packets

5. Try to write algorithm to create new meaningful knowledge

#0 define a knowledge packet
#1 define a target knowledge packet
#3 search for similar packets in the database (local or in network)
#4 order them to make a meaningful new packet or a packet sequence.
#5 visualize the new packet, so that people may get new ideas

---

Note : --- While working on projects like building gcc , chromium and other stuuf and
other projects , try to see to take a value from that and add to beautifulwork ---

--- may be consider each post of beautifulwork project as data and try to make algorithm work for all the posts. ---

--- may be collect different types of data from all over the world and also may be in different formats.

1. sysstat package has sadf command to make json file out of data file created by sar command. ---

--- even if there is nothing new in developing a software the logic may always have something new. It may be like that you are building several houses and should not get bored after building the first house because you can build a house using different way of logic. ---

--- May be one feature of beautifulwork is that people can document their command line sessions(like free software code) and share with others so that everyone can learn together.there should be analysis about the type of commands a person is using and about it's frequency and other things if possible. ---

--- may be a feature of beautifulwork is that it works like project hosting sites like github.com and savannah.gnu.org.
the key idea related is to share a improve the learning process.
commandline screenshot = a knowledge packet

* project hosted will be a commandline screenshot.
* commandline screenshot may have revisions.
* commandline screenshot may be stitched in a sequence to have a special meaning
* try to make data visualization of the packets.

--- may be you can use python for data analysis and other commandline application if needed in BW Project ---
--- may be you can use html and css for user interface and experiance ---
--- PHP for features of web application ---
--- Javascript for data visualization ---

--- may be put some things in the download part of BW project

* make a release for the /lib section and put a tarball.
* make a release for the /data section - may be as utilities for BW project.
* make a debian package for /code section - kind of code examples debian package ---

--- may be try to make a debian package of every directory in your github repository ---

--- may be try to make software for visualization of the programs in "code" section of beautifulwork repository. ---
--- may be try to make software tool for data analysis from "data" section for www.beautifulwork.org ---
--- may be try to make software library out of the "lib" section of beautifulwork repository ---

--- maybe you can use www.asciinema.org to make terminal videos and embed in www.beautifulwork.org ---