

# 2022第二次学习任务

---

本次任务分为：

- A. 理论基础学习部分1
- B. 实际操作部分
- C. 团队合作部分
- D. 理论基础学习部分2
- E. 能力提升

其中：任务A、B的截止时间为 11.1

任务D的截止时间为 11.11

任务C的截止时间为 11.20

任务E选择性完成

大家尽量在学有余力的情况下学习，如果感觉学习上有困难可以及时提出来

## A. 理论基础学习部分1

---

### 一. 学习Makefile

学习参考：

<http://c.biancheng.net/makefile/>

[https://blog.csdn.net/qg\\_34968572/article/details/102721670?ops\\_request\\_misc=%26request\\_id=%26biz\\_id=102&utm\\_term=Makefile&utm\\_medium=distribute.pc\\_search\\_result.none-task-blog-2~all~sobaiduweb~default-0-102721670.nonecase&spm=1018.2226.3001.4187](https://blog.csdn.net/qg_34968572/article/details/102721670?ops_request_misc=%26request_id=%26biz_id=102&utm_term=Makefile&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-102721670.nonecase&spm=1018.2226.3001.4187)

<https://blog.csdn.net/FRS2023/article/details/120839849?spm=1001.2014.3001.5506>

### 二. 学习OpenMP

群文件：



OpenMP入门.pdf

推荐视频：

[https://www.bilibili.com/video/BV1Ea411X78R/?spm\\_id\\_from=333.999.0.0&vd\\_source=d4d8725a1a30e189fa2cd9218fa9842a](https://www.bilibili.com/video/BV1Ea411X78R/?spm_id_from=333.999.0.0&vd_source=d4d8725a1a30e189fa2cd9218fa9842a)

[https://www.bilibili.com/video/BV1Pg411P7DW/?spm\\_id\\_from=333.999.0.0&vd\\_source=d4d8725a1a30e189fa2cd9218fa9842a](https://www.bilibili.com/video/BV1Pg411P7DW/?spm_id_from=333.999.0.0&vd_source=d4d8725a1a30e189fa2cd9218fa9842a)

## OpenMP API 用户指南:

[https://math.ecnu.edu.cn/~jypan/Teaching/ParaComp/books/OpenMP\\_sun10.pdf](https://math.ecnu.edu.cn/~jypan/Teaching/ParaComp/books/OpenMP_sun10.pdf)

## 推荐博客:

<https://zhuanlan.zhihu.com/p/51173703>

<https://blog.csdn.net/fengbingchun/article/details/15027507?spm=1001.2014.3001.5506>

<https://www.jianshu.com/p/9931c05f4058>

(以上仅代表个人观点, 仅给大家做个推荐, 建议实际学习过程自己去find)

## 完成超算习堂 - OpenMP 编程实训

[https://easyhpc.net/problem/programming\\_lab/2](https://easyhpc.net/problem/programming_lab/2)



### OpenMP 编程实训

OpenMP (Open Multi-Processing) 是一套支持跨平台共享内存方式的多线程并发的编程API, 使用C++和Fortran语言, 可以在大多数的处理器体系和操作系统中运行, 包括Solaris, AIX, HP-UX, GNU/Linux, Mac OS X 和Microsoft Windows, 包括一套编译器指令、库和一些能够影响运行行为的环境变量。OpenMP采用可移植的、可扩展的模型, 为程序员提供了一个简单而灵活的开发平台, 从标准桌面电脑到超级计算机的并行应用程序接口。

## 三. 学习并行计算基本概念

### 推荐书目:



(这本书比较精炼, 图书馆里可以借, 也可以自己去找电子pdf看)

## B. 实际操作部分

### 完成对HPL本机测试最优

#### 学习纲要:

1. 了解HPL.dat文件中每一行的含义。  
可参考<https://blog.csdn.net/gyx1549624673/article/details/86551466>  
(更多的可自行百度)
2. 了解HPL的调优手段, 发挥你们的搜索能力去了解吧。
3. 如果对于Linux系统基本指令不熟悉的可以接着学习与实践, 可以顺带了解一下Linux操作系统的原理。
4.  $G_{float} = \text{核心数(CPU几核)} \times \text{主频(GHz)} \times \text{每时钟周期浮点运算次数}$   
!!! 该公式用于计算理论峰值, 实际峰值为HPL跑出来的结果
5. 多进程运行命令 `mpirun -np 线程数 ./可执行文件`。例如: 4个进程运行xhpl, 指令为 `mpirun -np 4 ./xhpl`

友情提醒: 在测试调优的时候要保持电源连接, 后台少应用甚至无应用, 网络连接, 如果电脑有性能模式的话也可以打开, 不然有可能会产生较大的数值波动。还有一个就是在调参的时候不要一开始就把数值开太大了不然会跑很久。最后要测出自己的实际峰值与理论峰值的比值, 需要提交(最原始的比值, 以及调优最终的比值)

## C. 团队合作部分

团队合作建议**三人一组**完成, 大家自由组队

该部分内容较难, 大家需熟练掌握 **A. 理论基础学习部分** 再开始

### IPCC2021初赛赛题

1. 浏览关于IPCC优化培训的视频 (IPCC系列赛前培训6个视频)



赛前培训-IPCC讲师团



赛前培训-IPCC讲师团



赛前培训-IPCC讲师团



赛前培训-IPCC讲师团

<http://www.paratera-edu.org.cn/resource/index>

2. 下载2021IPCC初赛赛题并了解代码结构

SLIC算法介绍: <https://www.cnblogs.com/o2team/p/14928142.html>

3. 对程序进行热点分析, 确定主要优化函数
4. 对代码进行初步优化 (主要尝试利用所学的并行优化手段进行操作)
5. 记录优化过程

## D. 理论基础学习部分2

---

### 一. 学习MPI

群文件:



推荐资料03: MPI编程概述.pdf

推荐博客:

<https://blog.csdn.net/yjjgoodbay/article/details/110915852?spm=1001.2014.3001.5506>

<https://zhuanlan.zhihu.com/p/69497154>

<https://www.jianshu.com/p/2fd31665e816>

完成超算习堂 - MPI 编程实训

[https://easyhpc.net/problem/programming\\_lab/1](https://easyhpc.net/problem/programming_lab/1)



#### MPI 编程实训

MPI(Message Passing Interface)是一个跨语言的通讯协议,用于编写并行程序。与OpenMP并行程序不同, MPI是一种基于消息传递的并行编程技术。消息传递接口是一种编程接口标准,而不是一种具体的编程语言。简而言之, MPI标准定义了一组具有可移植性的编程接口。不同的厂商和组织遵循这个标准推出各自的实现,而不同的实现也会有其不同的特点。由于MPI提供了统一的编程接口,程序员只需要设计好并行算法,使用相应的MPI库就可以实现基于消息传递的并行计算。MPI支持多种操作系统,包括大多数的类UNIX和Windows系统。

### 二. 学习向量化操作

向量化指令查询:

<https://www.larurence.com/sse/#techs=AVX,AVX2>

推荐博客:

[https://blog.csdn.net/nbu\\_dahe/article/details/122157205?spm=1001.2014.3001.5506](https://blog.csdn.net/nbu_dahe/article/details/122157205?spm=1001.2014.3001.5506)

[https://blog.csdn.net/qg\\_36287943/article/details/105202721](https://blog.csdn.net/qg_36287943/article/details/105202721)

## E. 能力提升

学有余力和已有基础的同学，不强制要求完成

## 一. 算法学习

## 1. 掌握基础的DFS与BFS算法

经典题目：

## 排列数字

<https://www.acwing.com/problem/content/844/>

P2105 K皇后

<https://www.luogu.com.cn/problem/P2105>

## 走迷宫

<https://www.acwing.com/problem/content/846/>

## 2. 高阶：学习并行的DFS和BFS算法 (PDFS, PBFS)

可以在查找看看相关的论文，在 **任务C. IPCC2021初赛** 上可以运用上

## 二. 高性能计算学习路线

<https://heptagonhust.github.io/HPC-roadmap/>

## 系统性能分析快捷手册

## Linux Perf Analysis in 60s

- |     |                   |   |                       |
|-----|-------------------|---|-----------------------|
| 1.  | uptime            | → | load averages         |
| 2.  | dmesg -T   tail   | → | kernel errors         |
| 3.  | vmstat 1          | → | overall stats by time |
| 4.  | mpstat -P ALL 1   | → | CPU balance           |
| 5.  | pidstat 1         | → | process usage         |
| 6.  | iostat -xz 1      | → | disk I/O              |
| 7.  | free -m           | → | memory usage          |
| 8.  | sar -n DEV 1      | → | network I/O           |
| 9.  | sar -n TCP,ETCP 1 | → | TCP stats             |
| 10. | top               | → | check overview        |



