

دانشگاه گیلان

دانشکده فنی

گروه مهندسی کامپیوتر

گزارش کارآموزی

پیاده سازی سیستم توزیع شده با استفاده از MPI

نام دانشجو: مصطفی سن سبلی

شماره دانشجویی: ۹۲۱۲۲۶۲۱۱۹

تاريخ كارآموزي: تابستان ۱۳۹۷

استاد راهنما: دکتر حمیدرضا احمدی فر

سرپرست کارآموزی: مهندس جواد پور مصطفی

محل کارآموزی: دانشگاه گیلان، دانشکده فنی، آزمایشگاه نرم افزار

فهرست مطالب
فصل اول: معرفي محل كار آموزي
فصل دوم: کارهای انجام شده در دوره کارآموزی

مراجع:....

در این گزارش به شرح مفاهیمی از دستورات ترمینال سیستم عامل لینوکس پرداخته شده و در ادامه آن با استفاده از این دستورات، چندین سیستم کامپیوتری را در آزمایشگاه با استفاده از قطعات سخت افزاری، به همراه تعدادی از همکاران، شبکه کرده و برای هدف کلی این دوره که پیاده سازی یک سیستم توزیع شده با استفاده از مدل (MPI) Message Passing Interface به است، آماده کردیم، در نهایت، بعد از شبکه کردن و حل برخی از مشکلات و برداشتن بسیاری از موانع از قبیل مفاهیم مربوط به ssh و سرای مدیم.

در نهایت با نصب htop و دانلود چندین برنامه به زبان C که مخصوص پردازش موازی برنامه نویسی شده بودند به بهبود عملکرد کلی این سیستم ها پی بردیم.

فصل اول

تاريخچه دانشگاه گيلان

دانشگاه گیلان از سال ۱۳۴۶ با عنوان مدرسه عالی بازرگانی فعالیت آموزشی خود را آغاز نمود. و در سال ۱۳۵۳ به تصویب شورای گسترش آموزش عالی رسید و در سالهای ۵۵ – ۱۳۵۴ در چارچوب قرارداد بین دولتهای ایران و آلمان غربی سابق، دانشگاه گیلان ، تاسیس و از سال ۱۳۵۶ فعالیتهای آموزشی خود را با پذیرش ۱۵۵ دانشجو در ۹ رشته تحصیلی آغاز نمود .

در سال ۱۳۵۶ مدرسه عالی مدیریت و مدرسه عالی بازرگانی که تنها مراکز آموزش عالی در استان گیلان بودند با دانشگاه گیلان ادغام و فعالیت های آموزشی آن تا پایان سال تحصیلی ۱۳۵۸ با تعداد ۶۰۵ نفر دانشجو در ۱۴ رشته تحصیلی ادامه یافت.

پس از پیروزی انقلاب اسلامی، دانشگاه گیلان بطور مستقل و بدون وابستگی به آلمان، فعالیتهای آموزشی خود را در ۸ رشته تحصیلی و با حدود ۵۰۰ دانشجو از سرگرفت و با رشد و گسترش آموزش عالی در کشور، دانشکده های علوم پایه ، فنی،علوم کشاورزی،علوم انسانی و پزشکی ساخته شد و پس از آن نیز دانشکده تربیت بدنی و علوم ورزشی، دانشکده منابع طبیعی و معماری و هنر تاسیس شدند.

دانشگاه گیلان هم اکنون در ۶۴ رشته گرایش کارشناسی، ۱۶۶ رشته گرایش کارشناسی ارشد، ۹۳ رشته گرایش دکتری و در مجموع در ۳۲۳ رشته گرایش دانشجو می پذیرد.

این دانشگاه در حال حاضر با ۱۰ دانشکده و یک واحد پردیس و دو پژوهشکده(حوضه آبی دریای خزر و گیلان شناسی) با ۵۹۵ عضو هیأت علمی،حدود ۱۷۰۰۰ هزار دانشجو (حدود ۹ هزار دانشجوی دختر و ۷ هزار دانشجوی پسر) و به عنوان بزرگترین مراکز آموزش عالی در منطقه شمال کشور مشغول به فعالیت های آموزشی و پژوهشی است و سالانه حدود ۶۰۰۰ هزار نفر دانشجو را در ۴۹۴ رشته – گرایش در مقاطع تحصیلی کارشناسی، کارشناسی ارشد و دکتری برای دوره های روزانه ، شبانه ، پردیس و مجازی پذیرش می نماید و این دانشگاه با اعضای اتحادیه دانشگاه های دولتی کشور های حاشیه دریای خزر ، برخی از دانشگاه های کشورهای جنوب شرق آسیا و اتحادیه اروپا ارتباطات علمی و تبادل استاد و دانشجو دارد.

قرار گرفتن در جمع دانشگاه های برتر سه نظام بین المللی و رتبه بندی جهانی تایمز ،لیدن، شانگهای و دسته دانشگاه های یک درصد برتر دنیا و ده دانشگاه برتر جامع کشور جزئی از توانمندی های دانشگاه گیلان است.

درباره دانشکده فنی

دانشکده فنی در کیلومتر ۵ جاده رشت به قزوین و در پردیس اصلی دانشگاه قرار دارد. این دانشکده از سال ۱۳۵۷ فعالیت آموزشی خود را در ۳ رشته مکانیک، عمران و برق آغاز کرد.

دانشکده فنی در حال حاضر دارای تعداد ۱۱۲ عضو هیات علمی است که

- ۷ نفر دارای مرتبه استادی
- ۱۶ نفر دارای مرتبه دانشیاری
- ۶۹ نفر دارای مرتبه استادیاری

۲۰ نفر دارای مرتبه مربی می باشند و خدمات علمی را به طریق تدریس و پژوهش ارایه می دهند. تعداد کل دانشجویان این دانشکده در نیمسال اول ۹۵–۹۴ دارای ۳۳۵۶ نفر می باشد. از این تعداد ۸۲۹ نفر دانشجو در دوره های ارشد ، دکتری و ۲۳۴۶ نفر دانشجو در دوره کارشناسی روزانه و شبانه می باشد.

این دانشکده در حال حاضر دارای رشته گرایش های زیر است:

۳ رشته دکتری با گرایشهای:

مهندسی مکانیک: (مکانیک طراحی کاربردی)، (مکانیک – طراحی کاربردی دینامیک جامدات)، (مکانیک – طراحی کاربردی دینامیک – کنتل و ارتعاشات)، (مکانیک – تبدیل انرژی)

مهندسی عمران: (عمران – سازه)، (عمران – خاک ویی) مهندسی نساجی ۶ رشته کارشناسی ارشد با گرایشهای: مهندسی عمران: (سازه های دریایی، خاک پی، سازه، راه و ترابری) مهندسی مکانیک: (طراحی کاربردی ، تبدیل انرژی) مهندسي برق:(برق -قدرت)،(برق الكترونيك ديجيتال)،(برق الكترونيك ميكروالكترونيك)،(برق مخابرات-میدان)،(برق قدرت-سیستمای قدرت)،(برق قدرت ماشین های الکتریکی و الکترونیک قدرت)،(برق- مخابرات سيستم)،(برق-الكترونيك) مهندسی شیمی:(شیمی، شیمی فرآیندهای جداسازی) مهندسی نساجی: (شیمی نساجی و علوم الیاف) مهندسی کامپیوتر: (نرم افزار) ۶ رشته کارشناسی با گرایشهای: مهندسی مکانیک: (حرارت سیالات، طراحی جامدات، سیتم های دینامیکی و کنترل) مهندسی عمران: (عمران، عمران -عمران) مهندسی نساجی: (علوم الیاف) مهندسی شیـمی: (مهندسی شیمی، مهندسی شیمی–صنایع غذایی) مهندسي برق: (مهندسي برق، الكترونيك – قدرت، برق – قدرت) مهندسی کامپیوتر: (نرم افزار – سخت افزار) آدرس : رشت ، کیلومتر ۵ جاده رشت به قزوین، پردیس دانشگاه گیلان – دانشکده فنی صندوق پستی : ۳۷۵۶–۴۱۶۳۵ کدپستی : ۱۳۷۶۹–۱۳۹۶۶ دورنگار : ۳۳۶۹۰۲۷۱ – ۱۳۰ تلفنخانه : ۸- ۳۳۶۹۰۲۷۴ – ۱۳۰

معرفی گروہ مہندسی کامپیوتر:

گروه مهندسی کامپیوتر دانشگاه گیلان از مهر ماه ۱۳۸۶ با پذیرش ۴۰ دانشجو در یک رشته کارشناسی مهندسی کامپیوتر گرایش نرم افزار دوره روزانه و ۳ هیئت علمی و یک کارشناس فعالیت خود را آغاز نمود و در حال حاضر دارای رشته های کارشناسی مهندسی کامپیوتر گرایش های نرم افزار ، سخت افزار و کارشناسی ارشد مهندسی نرم افزار و کارشناسی ارشد معماری سیستم های کامپیوتری می باشد. در کارشناسی گرایش نرم افزار دارای ۱۹۷ دانشجو، در گرایش سخت افزار ۱۰۰ دانشجو و در رشته ارشد نرم افزار ۳۲ دانشجو و در کارشناسی ارشد معماری سیستم های کامپیوتری ۱۶ دانشجو مشغول به تحصیل هستند.

معرفی محل کار آموزی – آزمایشگاه نرم افزار

در رشته مهندسی کامپیوتر آزمایشگاه های نرم افزار و سیستم عامل وجود دارد و کار ما در آزمایشگاه نرم افزار انجام شد.

در آزمایشگاه نرم افزار تعداد ۱۷ کامپیوتر، سیم کشی های شبکه موجود بود. این آزمایشگاه در لابی دوم دانشکده فنی در طبقه اول قرار گرفته است.

فصل دوم

در ابتدا قبل از شروع هر کاری به مطالعه ی سیستم عامل لینوکس نسخه Ubuntu پرداختیم و اطلاعات علمی خود را ر این زمینه بالا بردیم.

در زیر به شرح مختصری از این کد ها می پردازیم:

Sudo

بسیاری از دستوراتی که در اینجا به توضیح آنها خواهیم پرداخت نیاز دارند که sudo به اول آنها اضافه شود. این دستور سطح دسترسی مدیریتی کاربر Root را به طور موقت برای ما فراهم می آورد که این سطح از دسترسی برای زمانی لازم است که ما که با دایرکتوری ها و یا فایل هایی که متعلق به حساب کاربری ما نیست کار می کنیم. به طور کلی در لینوکس یک کاربر مدیر به نام Root وجود دارد که معادل Root در ویندوز گروه administrator می باشد .این کاربر مدیر یا به عبارتی Super User قادر به انجام هر کاری می باشد در نتیجه انجام کارهای معمول روزانه بواسطه ی آن ممکن است خطر ناک باشد چرا که تایپ یک دستور آل یک کاربر تنها نیازمند دسترسی برای کار در دست اجرا می باشد .در برخی موارد استفاده از است اما در بیشتر اوقات کارها با استفاده از یوزر عادی پیش خواهد رفت .استفاده از Sudo ضروری نیازمند وارد کردن پسورد کاربر است.تنها کاربرانی با دسترسی Bub یا فی یا یا می دارد که معاده از ان یا یک موارد استفاده از یوزر عادی پیش خواهد رفت .استفاده از Sudo می فروری است اما در بیشتر اوقات کارها با استفاده از یوزر عادی پیش خواهد رفت .استفاده از Bub می با هر دستوری انترا این دستور می باشد .

دستورات مربوط به فایل و دایر کتوری

نماد تیلدا یا همان (~) برای دایر کتوری home در نظر گرفته شده است . اگر شما یک کاربر هستید پس ~ بیانگر مسیر home/user/ برای حساب کاربری شما می باشد.

- **Pwd** این دستور باعث می شود که دریابیم در حال حاضر در کدام دایرکتوری یا پوشه قرار داریم بنابراین اگر در سیستم گم شدیم و ندانیم در کدام بخش قرار داریم تایپ این دستور ما را از مکان فعلیمان مطلع خواهد ساخت .به عنوان مثال زمانی که در دایرکتوری desktop باشیم دستور خروجی "~/Desktop" را برای ما نمایش خواهد داد pwd .مخفف عبارت Print Working می باشد.
- LS این دستور لیست فایل های دایر کتوری جاری را نمایش می دهد . با استفاده از این اپشن می توان سایز فایل ها ، زمانی که فایل ساخته شده و سطوح دسترسی فایل را مشاهده کرد . یک نمونه از این دستور ~ ls می باشد که فایل هایی را که در دایر کتوری home قرار دارند نمایش می دهد.

- **Cd** دستور Cd اجازه تعویض دایر کتوری را به ما خواهد داد وقتی که یک ترمینال را باز کرده ایم در واقع در دایر کتوری home قرار داریم برای جابه جایی میان فایل سیستم ها باید از دستور Cd استفاده کنیم بعنوان مثال:
 - ۱. برای رفتن به دایر کتوری یا پوشه root از دستور "cd" استفاده می کنیم.
 - ۲. برای رفتن به پوشه ی home خود از دستور "cd" و یا "cd ~" استفاده می کنیم.
 - ۳. بر ای حرکت در سطوح دایرکتوری ها از دستور "**.. cd.** استفاده می کنیم.
 - ۴. برای برگشتن به دایرکتوری یا پوشه ی قبلی از دستور "- cd" استفاده می کنیم.
- ۸. برای رفتن به سطوح مختلفی از دایرکتوری به یکباره ، ابتدا باید مسیر کامل دایرکتوری را که قصد رجوع به آن را داریم ، مشخص کنیم .به عنوان مثال دستور "cd ~/Desktop"
 موجب انتقال کاربر ما به زیر پوشه ی Desktop که دورن پوشه ی home قرار دارد ، می شود.
- **Cp** دستور Cp یک کپی از فایل برای ما می سازد به عنوان مثال دستور "cp file itpro" دقیقا یک کپی از محتوای فایل با نام itpro ایجاد کرده اما فایل "file" همچنان وجود خواهد داشت . برای کپی کردن دایرکتوری ها نیز باید از دستور "r- cp" استفاده کرد برای نمونه دستور r- cp" itpro یک کپی کردن دایرکتوری ها نیز باید از دستور "r- cp" استفاده کرد برای نمونه دستور r- cp" توجاد خواهد کرد . باید در نظر داشته باشیم که کپی دایرکتوری ها به گونه ای است که از پوشه و ایجاد خواهد کرد . باید در نظر داشته باشیم که کپی دایرکتوری ها به گونه ای است که از پوشه و تمامی فایلها و زیر پوشه و همچنین از تمامی محتویات زیر پوشه ها کپی می گیرد و این سلسله مراتب کپی برداری از محتوای دایرکتوری ها در استفاده از این دستور حفظ خواهد شد.اصطلاحا در زبان انگلیسی این ساختار از دایرکتوری را recursively یا بازگشتی گویند که r نیز در دستور ذکر شده به این کلمه اشاره می کند.
- دستور mv یک فایل را به مکانی دیگر منتقل کرده و یا یک فایل را تغییر نام می دهد .مثلا دستور "itpro" یک فایل موجود با نام "file" را به "itpro" تغییر نام می دهد . اما دستور "itpro" فایل موجود با نام "tipro" را به دایرکتوری desktop بدون تغییر دادن نام آن . انتقال می دهد.
 - برای کوتاه شدن دستور همانند مثال بالا می توان از نما (~) به جای پوشه ی home
 استفاده کرد.
- ۲. باید توجه داشت که زمانی که از دستور mv به همراه sudo استفاده می کنیم نیز می توانیم
 از علامت (~) به جای دایر کتوری home استفاده کنیم چرا که terminal به پوشه
 home مربوط به کاربر ما اشاره خواهد داشت. با این حال زمانی که یک root shell را با

استفاده از دستور i- sudo یا sudo -s باز می کنیم ، نماد (~) به پوشه home مربوط به اکانت root اشاره خواهد کرد که در واقع پوشه home کاربر ما نیست.

- **Rm** دستور ۲m برای حذف یک فایل از یک دایر کتوری استفاده می شود rm . مخفف عبارت removeاست.
- Rmdir عبارت dir مخفف کلمه ی directory می باشد بنابراین می توان دریافت که دستور rmdir به حذف یک دایر کتوری و تمام محتویات درون rmdir به حذف یک دایر کتوری خالی می پردازد . برای حذف یک دایر کتوری و تمام محتویات درون به جای rm از دستور r- rm استفاده می شود .همانطور که می دانیم r نیز بیانگر عبارت recursivelyمی باشد.
 - Mkdir به کمک دستور mkdir قادر خواهیم بود که یک دایرکتوری ایجاد کنیم . به عنوان مثال دستور "mkdir itpro" یک پوشه یا دایرکتوری با نام itpro ایجاد خواهد کرد mk . مخفف عبارت make می باشد.
 - این دستور کلیه ی مطالب و دستورات تایپ شده در صفحه ی ترمینال را پاک می کند.
- Man بواسطه این دستور می توان به راهنمای سایر دستورات دسترسی پیدا کرد.مثلا دستور man" "manصفحه ای از اطلاعات مرتبط با خود دستور man را نمایش می دهد . این دستور صفحه ای پر از نوشته و اطلاعات را در اختیار قرار می دهد که ممکن است کمی گیج کننده باشد اما بخشی که باید به آن توجه داشته باشیم توضیحات دستور و کاری که انجام می دهد، می باشد. برای scrol کردن به پایین این صفحه ی راهنما از کلید space و برای خروج از آن از کلید q استفاده می کنیم. در ادامه بیشتر به بررسی دستور man خواهیم پرداخت...
- **Sudo** این دستور برای انجام عملیاتی روی فایل ها استفاده می شود که تنها کاربر root قادر به اعمال این تغییرات است.مثلا عملیات باز گردانی یکی از داکیومنت هایی که به طور تصادفی به کابر دیگری منتقل شده،به پوشه دایر کتوری کابر خودمان نیازمند دسترسی root می باشد. به طور معمول برای انتقال یک فایل از دستور **mv //mydoc.odt**

Sudo ما نیست ...برای این منظور می توانیم از دستور قادر به ویرایش و تغییر وضعیت فایل در خارج از پوشه ی home ما نیست .برای این منظور می توانیم از دستور **mv //mydoc.odt // Documents//mydoc.odt** استفاده کنیم این دستور فایل را با موفقیت به مکان مورد نظر بازگردانی و منتقل خواهد کرد.برای انجام چنین عملیاتی با کمک این دستور اگرچه کاربر ما استاندارد نیست اما سطوح دسترسی آن کمتر از یک administrator نیز نخواهد بود.

اجرای فایل در یک دایر کتوری:

اگر قصد اجرای یک فایل در دایرکتوری را به کمک خط فرمان داشته باشیم می توانیم از دستور "./filenam.extension" استفاده کنیم .که به جای filename نام فایل مورد نظر را تایپ می کنیم و extension دستور کاربر را قادر می سازد تا فایل مورد نظر ما می باشد. پس از هدایت شدن به سوی دایرکتوری فایل ، این دستور کاربر را قادر می سازد تا فایل کامپایل شده بواسطه ی GCC و یا هر کامپایلر مربوط به زبان های برنامه نویسی دیگر را اجرا کند. دستور عنوان شده به پسوند فایل اشاره دارد و باید توجه داشت که متفاوت با سایر سیستم عامل های لینوکسی دیگر ، پسوند فایل در اوبونتو اهمیت ندارد و در واقع extension یا همان پسوند بسته به زبانی که کد منبع با آن نوشته شده متفاوت خواهد بود . همچنین این ممکن نیست که برای زبانهای کامپایل شده مثل C و ++C که Source code یا کد منبع را به طور مستقیم اجرا کنند و فایل ابتدا باید کامپایل شده مثل C و ++C که PC یا Source code یا کد منبع را به طور مستقیم اجرا کنند و فایل ابتدا باید کامپایل شود بدین معنی که آن باید ابتدا از یک زبان برنامه نویسی قابل خواندن توسط انسان به زبانی قابل درک برای کامپیوتر ترجمه شود. از پسوندهای موجود می توان به ".C" برای کد منبع زبان C و "rop" می می فایل خواندن توسط انسان به زبانی قابل مرای C ++ و "dr" برای Ruby و "rop" برای موجود می توان به ".C" برای کد منبع زبان C و "rop" مورتی این می می می می می می می می در ستوی این می و می توان . مرای C ++ و "dr" برای Ruby و "rop" برای Ruby و... اشاره کرد. در نهایت فایل تنها در صورتی اجرا می شود که اگر سطوح دسترسی ویا Permission ها برای دستیایی به فایل درست باشند .

دستورات مربوط به دستیابی به اطلاعات سیستم

Df دستور df فضای مورد استفاده فایل سیستم برای همه ی پارتیشن های نصب شده را نشان می
 دهد . اما استفاده از دستور "df -h" احتمالا مفید تر خواهد بود چرا که این دستور در گزارش خود
 از (M) یا مگا بایت و از (G) یا گیگابایت به جای Block ها استفاده می کند -h نیز بیانگر عبارت
 "human readable" می باشد.

😣 🖨 🗊 🖉 mram	ezani@mrame	zani-virtua	l-machine: ~		
mramezani@mra	mezani-vir	tual-mach	ine ~\$ df		
Filesystem	1K-blocks	s Used	Available	Use%	Mounted on
/dev/sda1	19857450	5 2480464	16380896	14%	/
udev	505124	1 4	505120	1%	/dev
tmpfs	204960	784	204176	1%	/run
none	5120	0 0	5120	0%	/run/lock
none	512390	5 76	512320	1%	/run/shm
/dev/sr0	715052	2 715052	0	100%	/media/Ubuntu 12.04 L
TS amd64					
mramezani@mra	mezani-vir	tual-mach	ine:~\$ df	-h	
Filesystem	Size Us	sed Avail	Use% Moun	ted or	ר <u>ר</u>
/dev/sda1	19G 2.	4G 16G	14% /		
udev	494M 4	0K 494M	1% /dev		
tmpfs	201M 78	34K 200M	1% /run		
none	5.0M	0 5.0M	0% /run	/lock	
none	501M	76K 501M	1% /run	/shm	
/dev/sr0	699M 69	99M 0	100% /med	ia/Ubu	untu 12.04 LTS amd64
mramezani@mra	merani-Kig	wal-mach	من تكمن عرا	ا ـ اند	TPro.ir

 Du این دستور میزان دیسک استفاده شده برای دایر کتوری ها را نمایش می دهد . این دستور همچنین می تواند فضای مورد استفاده برای تمامی زیر پوشه ها و برای مجموع تمام دایر کتوری هایی که انها را اجرا کرده ایم را نمایش دهد. در تصویر زیر مثالی برای این دستور نمایش داده شده است . که در آن -S به معنی summary و -h به معنی human readable است.

	mramezani@mramezani-virtual-machine: ~
mramezai	ni@mramezani-virtual-machine: <mark>~</mark> \$ du /media /floppy
1592	/media/20140510_1140/Power ISO 47
1594	/media/20140510_1140
4	/media/floppy0
1602	/media
mramezar	ni@mramezani-virtual-machine: <mark>~</mark> \$ du -sh /media /floppy
1.6M	/media
mramezai	ni@mramezani-wirtual-machine:~S

- Free این دستور مقدار فضای آزاد و استفاده شده ی memory را در سیستم نمایش می دهد و دستور "free -m" این اطلاعات را بر حسب مگابایت به ما نشان خواهد داد که این دستور کارایی بیشتری نسبت به دستور اول دارد.
- Top این دستور مخفف عبارت ('table of processes') می باشد که اطلاعات روی سیستم Top این دستور مخفف عبارت ('table of processes') می باشد که اطلاعات روی سیستم لینوکس ما ، پردازش های در حال اجرا ، و منابع سیستم از جمله CPU و RAM و مجموع برنامه های در حال اجرا را نشان می دهد . برای خروج از این دستور top از کلید "q" استفاده می کنیم.
- Uname استفاده از دستور uname به همراه آپشن -a تمامی اطلاعات سیستم شامل نام دستگاه ، نام هسته و سایر جزئیات دیگر را نمایش می دهد . بیشترین کاربرد این دستور برای چک کردن kernel ای است که استفاده می کنیم.
 - ا ip addr دستور **Ip addr** گزارشی از کارت شبکه های روی سیستم را نمایش می دهد.
 - Lsb_release دستور Isb_release به همراه آپشن -a اطلاعات مربوط به ورژن لینوکسی که استفاده می کنیم را نمایش می دهد . بعنوان مثال:

mramezani@mramezani-virtual-machine: ~
mramezani@mramezani-virtual-machine: ~\$ lsb release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 12.04 LTS
Release: 12.04
Codename: precise
mramezani@mramezani-virtual-machine:~\$

ایجاد کردن یک کاربر جدید:

به کمک دستور "adduser newuser" می توان یک حساب کاربری جدید ایجاد کرد ، باید در نظر داشته باشیم که این دستور با سطح دسترسی کاربر Root اجرا خواهد شد و کاربر ریشه است که می تواند یوزر و یا گروه اضافه کند. بعد از تایپ دستور "sudo adduser newuser" سیستم از ما پسورد حساب کاربریمان را برای احراز هویت درخواست می کند و پس از آن یوزر ایجاد خواهد شد و دستورات مرحله به مرحله پیش می روند و از ما خواسته می شود که پسورد یوزر جدید را وارد کنیم و اطلاعات جانبی دیگری ازجمله نام کامل : (ITpro)، شماره اتاق ، شماره تلفن از ما پرسیده می شود و در نهایت باید صحت یا عدم درستی اطلاعات وارد شده را با استفاده از کلید های Y و یا n مشخص کنیم . بدین ترتیب حساب کاربری جدید ایجاد خواهد شد .

آپشن ها:

گاه ممکن است که دستورات پیش فرض با افزودن آپشن هایی به آنها تغییر کرده و ویرایش شوند. مثلا استفاده از آپشن -Sبه همراه دستور ls که به صورت s-s تایپ می شود اندازه ی فایل در این لیست را بر می گرداند. همچنین گزینه ی -h باعث می شود که سایز فایل ها در لیست با فرمت قابل درک برای انسان human) (human نمی توان ای در می شود که سایز فایل ها در لیست با فرمت قابل درک برای انسان human) مثال می توان به دستور "s-sh" اشاره کرد که در واقع همان وظیفه ی دستور h-s-sh" را انجام می دهد.بسیاری از آپشن هایی که نسخه ای طولانی دارند با پیشوندی از دو علامت خط تیره یا dash به جای یک خط تیره به کار می روند .بنابراین "s-size - human --size -- size -- size -- human به مان ستور ha می باشد .

©©© mrameza	ni@mramezani-virtual-	machine: ~
mramezani@mrame total 48K	zani-virtual-machi	ne: <mark>~\$ls-s-h</mark>
4.0K Desktop	12K examples.des	<pre>xtop 4.0K Pictures 4.0K Videos</pre>
4.0K Documents	4.0K itpro	4.0K Public
4.0K Downloads	4.0K Music	4.0K Templates
mramezani@mrame total 48K	zani-virtual-machin	ne: <mark>~\$ lssizehuman-readable</mark>
4.0K Desktop	12K examples.des	ctop 4.0K Pictures 4.0K Videos
4.0K Documents	4.0K itpro	4.0K Public
4.0K Downloads	4.0K Music	4.0K Templates
mramezani@mrame	zani-victual-machin	ITPro ir

دستورات man و info و info از مهم ترین ابزارهای خط فرمان می باشند . تقریبا تمامی دستورات و برنامه ها در لینوکس یک man یا manual فایل دارند .بنابراین پیدا کردن این فایل ها با تایپ دستور "" man "command" به آسانی صورت گرفته و یک راهنمای طولانی را برای دستور ذکر شده در اختیار ما قرار می دهد . برای مثال دستور "man mv" راهنمای مربوط به دستور mv را برای ما نمایش می دهد . می دانیم که mv مخفف عبارت move می باشد. برای حرکت به بالا و پایین لیست راهنمای آورده شده با دستور man می توان از کلید gace و یا کلید های جهت دار استفاده کرد .همچنین برای بازگشتن به خط فرمان کلید q را می زنیم . دستور "man man" راهنمای مرتبط با دستور man می بازگشتن به خط دستور مناسبترین گزینه و راهنما برای شروع کار با دستورات است .

دستور "man intro" بسیار مفید است و این دستور به معرفی و نمایش راهنمای دستورات کاربر می پردازد که به خوبی نوشته شده و در واقع به گونه ای مختصر دستورات لینوکس را شرح داده است .عملا استفاده از این دستور خوانندگان را از مطالعه ی این مقاله بی نیاز می کند . D:

همچنین علاوه بر صفحات manual صفحات info نیز وجود دارد که دستور "info info" به معرفی صفحات info می پردازد. برخی از توسعه دهندگان نرم افزار ها info را به man ترجیح می دهند مثل . GNUبنابراین اگر از دستور و یا برنامه ای استفاده کردیم که صفحه ی man نداشت باید برای دستیابی به

راهنمای آن دستور info را چک کنیم .

عملا تمامی دستورات ، آپشن -h یا --help که برای یک تعریف کوتاه و کاربردی از دستور و آپشن های آن تولید شده را درک می کنند.به عنوان مثال برای مشاهده ی عملکرد این آپشن می توان دستور "man -h" و یا "man - help" را امتحان کرد . امکان دارد که یک برنامه قادر به درک معنی آپشن -h نباشد به همین دلیل باید ابتدا صفحه ی man و یا info را بررسی کرده و گزینه --help را به جای -h کار برد.البته این احتمال معمولا نادر است ...

دستورات مربوط به مراحل شبکه کردن:

Ifconfig

در سیستم های یونیکس ، لینوکس و مکینتاش از دستور ifconfig به جای ipconfig باید استفاده کنید. برای اینکه صفحه help مربوط به این دستور را نگاه کنید باید از دستور ifconfig -help استفاده کنید. خروجی زیر پس از استفاده تنها از دستور ifconfig حاصل شده است.

eth0 Link encap:Ethernet HWaddr 00:60:08:17:63:A0
inet addr:192.168.1.101 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING MTU:1500 Metric:1
RX packets:911 errors:0 dropped:0 overruns:0 frame:0
TX packets:804 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
Interrupt:5 Base address:0xe400

lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0

UP LOOPBACK RUNNING MTU:3924 Metric:1

RX packets:18 errors:0 dropped:0 overruns:0 frame:0

TX packets:18 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

دستور ifconfig آدرس IP، subnet mask و default gateway را برای کارت شبکه های محلی و loopback نشان می دهد.

دستور pico

این دستور یک ویرایشگر متن ساده است که برای باز کردن و ویرایش فایل های متنی بکار می رود. برای مثال برای تغییر در تنظیمات شبکه یک سیستم کامپیوتری می توانیم از کد دستری زیر استفاده کنیم: Sudo pico /etc/network/interfaces این قطعه دستور، فایل متنی interfaces در زیر پوشه network در پوشه ی etc را باز می کند. پس از اعمال تغییرات می توان با دکمه های ترکیبی ctrl+x فایل را ذخیره کرده و خارج شوید.

sudo service networking restart دستور

این دستور باعث می شود که تنظیماتی که در رابطه با شبکه تغییر داده شد، به روز رسانی و قابل استفاده شود.

دستور ping

با وارد کردن این دستور در یک سیستم به صورت برای مثال ping 192.1685.100.100 در می یابیم که آیا این سیستم با سیستم با آدرس ۱۹۲٬۱۶۸٬۱۰۰٬۱۰۰ ارتباط برقرار کند یا خیر.

کد های مربوط به Secure Shell (ssh)

دستور ssh

عموما این دستور به همراه گزینه های دیگر همراه است، که این گزینه مربوط به سیستم ها دیگر است. برای مثال : ssh (user name)@(IP Address)

دستور sudo apt-get install

از این دستور می توان برای دانلود و از نصب هر برنامه ای استفاده کرد.

دستور ssh-keygen

از این دستور برای تولید زوج ssh-keyها برای اتصال سرور به کلاینت ها استفاده می شود .این key pair ها را می توان با رمزگزاری rsa تولید کرد که یک گزینه برای آن می باشد. وقتی که این کار انجام شد دو فایل در پوشه ی پیش فرض ssh./ تولید می شود. یکی id-rsa.pub و دیگری id-rsa.pub.

کد مربوط به به روز رسانی سیستم

کد sudo apt-get update

با این کد سیستم و تمامی نرم افزار های آن را به آخرین نسخه خود به روز رسانی میکنیم.

کد chown

از این دستور برای تغییر مالکیت یک فایل می توان استفاده کرد.

مراحل انجام کار در آزمایشگاه:

در جلسه اول ابتدا با این مسئله روبرو بودیم که با چه روشی سیستم ها را برای پردازش موازی پیاده سازی کنیم:

- HPC .۱
- MPI .۲

که در زیر به هر یک از آنها بصورت مختصر اشاره می کنیم.

HPCمحاسبات با کارایی بالا

HPCمحاسبات با کارایی بالا استفاده از پردازش موازی برای اجرای برنامه های کاربردی پیشرفته کارآمد قابل اعتماد و سریع است.

HPCبه ویژه برای سیستم هایی که بیش از ۱۰۱۲ عملیات نقطه شناور در هر ثانیه عمل می کند، اعمال می شود.

HPC به عنوان یک اصطلاح گاهی اوقات نیز به عنوان مترادف برای ابر رایانه استفاده می شود، اگرچه از نظر تکنیکی ابر کامپیوتر یک سیستم است که در حال حاضر در بالاترین میزان عملیات رایانه ها یا نزدیک آن قرار می گیرد. برخی ازاین ابر رایانه ها در بیش از یک عملیات (پتافلپ) یا ۱۰۱۵ عملیات شناور در هر ثانیه کار می کنند.

اکثر کاربران سیستم های HPC عبارت اند از: محققان، مهندسان و موسسات علمی ، بعضی از سازمان های دولتی

به ویژه ارتش، برای HPC به برنامه های پیچیده نیز متکی هستند.

سیستم های با کارایی بالا اغلب از اجزای سفارشی علاوه بر اجزای به اصطلاحTOP) (استفاده می کنند. *HPC* علاقه مند به تمامی کسب و کارها درهمه ابعاد ، مخصوصاً برای پردازش معاملات و انبارهای داده است. HPC،اساساً یک شبکه گره است که هر کدام شامل یک یا چند تراشه پردازش، و همچنین حافظه خاص خود است.

محاسبات موازى

برنامه های سیستم های HPC باید به "برنامه های" کوچکتر تقسیم شوند که موضوعات آنها مربوط به یک هسته است.

هسته ها باید بتوانند با همدیگر برنامه های بزرگتری را بسازند و بتوانند به طور موثر با یکدیگر ارتباط برقرار کنن

همچنین سیستم به عنوان یک کل باید سازماندهی خوبی داشته باشد.

برنامه های سیستم های HPC مقدار زیادی از اطلاعات را می سازد که می تواند برای سیستم های فایل استاندارد

همچنین سخت افزار ذخیره سازی برای مقابله با آن بسیار دشوار باشد.

سیستم های فایل استاندارد که برای استفاده شخصی تعریف شده اند ممکن است محدودیتی در اندازه فایل، تعداد

فایل ها یا ذخیره سازی کامل داشته باشند.

سیستم های فایل های HPC باید قادر باشند رشد کنند تا مقدار زیادی از داده ها را انتقال دهند و سریعاً انتقال یابند.

محققان علاوه بر داده های مورد استفاده فعلی، اطلاعات قبلی را برای مقایسه یا به عنوان نقطه شروع برای پروژه های آینده نگه می دارند. داده های قدیمی تر در سیستم های ذخیره سازی بایگانی نگهداری می شوند Kraken، برای مثال، از یک سیستم ذخیره سازی نوار مغناطیسی استفاده می کند که می تواند چندین پتابایت (میلیون ها گیگابایت) داده را ذخیره کند. MPI:

رابط فرستادن پیام یا Message Passing Interface به صورت اختصاری MPI ، یک سیستم استاندارد فرستادن پیام است که توسط گروهی از محققان دانشگاهی و صنعتی طراحی شده است. این رابط روی طیف وسیعی از رایانه های موازی پیاده سازی شده است.

استاندارد MPI ساختار و کاربرد چند رویه اساسی را تعریف می کند که برای ساختن برنامه های موازی که از فرستادن پیام استفاده می شوند، از آن ها می تواند استفاده شود. این رویه ها برای تعداد زیادی از کاربران قابل استفاده هستند و علاوه بر آن ماهیت استاندارد این رویه ها باعث می شود نرمافزار های نوشته شده با این رویه های کتابخانه ای قابلیت حمل بالایی روی سکو های مختلف داشته باشند.

پیادهسازیهای مختلفی از MPI صورت گرفتهاست که بسیاری از آنها کاملاً تستشده و مطمئن هستند. بسیاری از آنها آزاد و حتی بعضی بدون محدودیت و توافقنامه قابل استفاده هستند. پیادهسازیهای MPI برای زبانهای مختلفی از جمله فرترن، زبانهای C و++C ، جاوا و غیره موجود است. دو پیادهسازی معروف عبارتند از Open MPIوMPIL.

کتابخانههای استانداردی مانند MPI و PVM صنعت توسعه نرمافزارهای موازی را رونق دادهاند و ساختن نرمافزارهای موازی و مقیاس پذیر بزرگ را امکان پذیر ساختهاند.

و در آخر به این نتیجه رسیدیم که از روش MPI استفاده کنیم.

در گام بعدی آمدیم و سیستم های داخل های آزمایشگاه را چک کردیم. دیدیم که برخی از آنها اصلا روشن هم نمی شوند برخی دیگر هم سیستم سخت افزاری یا نرم افزاری شان مشکل داشت. و در نهایت ۵ سیستم را انتخاب کردیم که به این ترتیب بودند:

سیستم شماره ۴۲

سیستم شماره ۴۸

سیستم شماره ۴۹

سیستم شماره ۶۰

سیستم شماره ۶۳

برای سهولت و راحتی کار خودمان در ادامه آمدیم و روی این سیستم ها، سیستم عامل Ubuntu <u>ر</u>ا نصب کردیم و سپس یک User دیگر در آن سیستم عامل ها ایجاد کردیم به نام mpiuser و از آن روز به بعد کارمان را در این user ادامه دادیم. برای تمامی سیستم ها رمزی را در نظر گرفتیم که guilanmpi بود. سپس در ادامه یک سوییچ خریداری و تهیه کردیم که بتوانیم این ۵ سیستم را بصورت LAN به هم دیگر شبکه

سپس در ادامه یک سوییچ حریداری و تهیه کردیم که بتوانیم این ۵ سیستم را بصورت LAN به هم دیگر شبکه کنیم.

سپس IP سیستم ها را از طریق ترمینال سیستم عامل Ubuntu شناسایی و تغییر دادیم: که مراحل کار آن بصورت زیر بود:

با زدن دکمه های ترکیبی CTRL+ALT+T می توان وارد پنجره ترمینال شد.

یک راه ساده دیگر برای باز کردن ترمینال این است که کلیک راست کرده و گزینه Terminal را انتخاب کنیم.

وقتی که این کار را انجام دادیم و پنجره ترمینال باز شد، مانند شکل زیر:



می توان با وارد کردن دستور ifconfig، وارد مشخصات شبکه ای سیستم مان شویم که با صفحه مانند شکل زیر مواجه می شویم:

😣 🖻 💷 a@a-virtual-machine: ~ a@a-virtual-machine:~\$ ifconfig ens33 Link encap:Ethernet HWaddr 00:0c:29:85:88:7b inet addr:192.168.126.137 Bcast:192.168.126.255 Mask:255.255.255.0 inet6 addr: fe80::20c:29ff:fe85:887b/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:109632 errors:0 dropped:0 overruns:0 frame:0 TX packets:42078 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:146543985 (146.5 MB) TX bytes:2692863 (2.6 MB) 10 Link encap:Local Loopback inet addr:127.0.0.1 Mask:255.0.0.0 inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1 RX packets:204 errors:0 dropped:0 overruns:0 frame:0 TX packets:204 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:15526 (15.5 KB) TX bytes:15526 (15.5 KB) a@a-virtual-machine:~\$

همانطور که در عکس میبینید ۲ تا واسط کاربری شبکه را برای شما نمایش میدهد.

ens33: که نام اینترفیس شبکه شما می باشد که با تغییر استاتیک در این قسمت می توانید با چند سیستم دیگر به صورت LAN ارتباط برقرار کنید یا با تغییر dhcp می توانید به شبکه اینترنت وصل شوید. lo : این نیز به نوعی اینترفیس شبکه می باشد ولی این در همه سیستم ها موجود می باشد و با آدرس ۱۲۷,۰,۰,۱

شما یک وب سرور نصب شده باشد و آن را اجرا کنید و می خواهید با استفاده از کروم یا فایر فاکس آن را جست و جو کنید . برای این کار سیستم از این اینترفیس برای جست و جو های محلی استفاده خواهد کرد. در ادامه:

کاری که ما کردیم این بود که آمدیم و آدرس هایی که به طور پیش فرض در قسمت اینترفیس برای ens33 قرار داده شده بود را به طور دلخواه خودمان تغییر دادیم.

برای اینکار با وارد کردن دستور : sudo pico /etc/network/interfaces وارد صفحه ای مانند شکل زیر می شویم که قبل از آن از ما در خواست رمز عبور می شود که همان رمزی است که ما در هنگام ایجاد یک user جدید تعیین کردیم:



همانطور که در این عکس مشاهده می کنیم، اینترفیس شبکه بصورت dhcp تنظیم شده. کاری که می کنیم این است که آن را بصورت استاتیک تنظیم می کنیم و این کار را برای هر یک از ۵ سیستم انجام می دهیم.

نتیجه نهایی مانند شکل زیر است:



برای تفسیر کردن تغییراتی که ایجاد کردیم میتوان گفت که:

اینترفیس شبکه را از حالت dhcp به static تغییر دادیم و سپس برای این سیستم یک آی پی آدرس و یک نت ماسک تعریف کردیم.

توجه داشته باشیم که هر یک از سیستم ها <u>آی پی آدرس متفاوتی</u> با بقیه دارد ولی <u>نت</u> ماسک ها یکسان است.

در ادامه برای اینکه بتوانیم تغییرات ایجاد شده را ذخیره کنیم و از این پنجره بیرون بیایم از دکمه های ترکیبی ctrl+x استفاده میکنیم که از ما میپرسد که این تغییرات را با همین نام در همین فایل ذخیره کنیم یا خیر. برای ادامه کار با فشردن دکمه ی Y و سپس زدن دکمه ی ENTER می توانیم تغییرات را ذخیره کرده و از این پنجره خارج شویم.

اسم	IP Address	شماره سیستم
Master	192.168.1.10	42
Slave1	192.168.1.11	48
Slave2	192.168.1.12	49

در جدول زیر شماره سیستم ها و آدرس آی پی آن ها درج شده است:

Slave3	192.168.1.13	60
Slave4	192.168.1.14	63

حال برای اینکه این تغییرات ما بطور کلی در سیستم اعمال شود باید از دستور زیر استفاده کنیم تا تنظیمات اعمال شده ریستارت شود:

sudo service networking restart

بعد از این کار برای حصول اطمینان از اینکه آیا سیستم ها به هم متصل بودند یا خیر از دستور ping استفاده کردیم به این صورت که در هر یک از سیستم ها چک کردیم که آیا با تک تک سیستم های دیگر در ارتباط است و packet ارسال می شود یا خیر.

برای مثال در سیستم شماره ۴۲ یا همان master بصورت شکل زیر از اتصال master به سیستم های دیگر مطمئن شدیم:

😣 🖨 💼 🛛 a@a-virtual-machine: ~
a@a-virtual-machine:~\$ ping 192.168.100.101 PING 192.168.100.101 (192.168.100.101) 56(84) bytes of data. 64 bytes from 192.168.100.101: icmp_seq=1 ttl=64 time=0.235 ms 64 bytes from 192.168.100.101: icmp_seq=2 ttl=64 time=0.301 ms 64 bytes from 192.168.100.101: icmp_seq=3 ttl=64 time=0.239 ms
64 bytes from 192.168.100.101: icmp_seq=4 ttl=64 time=0.221 ms 64 bytes from 192.168.100.101: icmp_seq=5 ttl=64 time=0.248 ms ^Z
[5]+ Stopped ping 192.168.100.101
a@a-virtual-machine:~\$ ping 192.168.100.102
64 hytes from 192 168 100 102; icmp sed=1 ttl=64 time=0.694 ms
64 bytes from 192.168.100.102: icmp_seq=1 ttl=64 time=1.06 ms
64 bytes from 192.168.100.102: icmp seg=3 ttl=64 time=1.13 ms
64 bytes from 192.168.100.102: icmp_seq=4 ttl=64 time=2.01 ms ^Z
[6]+ Stopped ping 192.168.100.102
a@a-virtual-machine:~\$

همانطور که در تصویر میبینید master در حال ارسال packet به گره (سیستم) های دیگر میباشد، این یعنی ارتباط برقرار است. در ادامه در فایل Hosts هر یک از سیستم ها در پوشه etc در computer آی پی تمامی سیستم ها را در تک تک سیستم ها وارد کردیم و برای آن ها نامی انتخاب کردیم که در ادامه برای سهولت کار انجام شد .روند کار به این صورت بود:

در پنجره ترمینال دستور sudo nautilus را وارد کردیم و سپس سیستم از ما رمز عبور خواست، آن را وارد کردیم و سپس بصورت یک super user وارد محیط گرافیکی کامپیوتر شدیم، در قسمت چپ وارد computer شده و در آنجا در پوشه ی etc به دنبال فایل hosts گشتیم، وارد آن شدیم . آن را به این صورت تغییر دادیم:



و در آخر هم گزینه save را زده و از محیط گرافیکی خارج شده و دوباره وارد ترمینال شدیم.

بعد از اتمام این کار ها نیاز داشتیم که بتوانیم از طریق سیستم master وارد تک تک سیستم ها بشویم. برای این کار از sshکمک گرفتیم. در ابتدانیاز داشتیم که به اینترنت وصل شویم و openssh را در master دانلود و نصب کنیم. لذا چون سوویچ به اندازه ۵ عدد سوکت داشت، یکی از سیستم ها را از سوکت کنده و یک لپ تاپ را به شبکه وصل کردیم. سپس با استفاده از اینترنت گوشی که به صورت هات اسپات از آن در لپتاپ استفاده میکردیم، لپ تاپ را با گوشی بریج (Bridge) کردیم تا تمامی سیستم های متصل به شبکه بتوانند از اینترنت آن استفاده کنند، سپس تنظیمات شبکه سیستم ها را که با استفاده از دستوری که در بالا ذکر کردیم دوباره به حالت dhcp باز گرداندیم و با استفاده از مرور گر فایرفاکس چک کردیم که آیا در حال حاضر به اینترنت دسترسی داریم یا خیر.

Sudo apt-get install open-ssh server

مدتی طول کشید و عملیات نصب open ssh روی سیستم ها به اتمام رسید.

سپس از دستور زیر برای ورود به سیستم های دیگر استفاده کردیم:

Ssh username@IP_Addres

در شكل زير ميبينيد كه از طريق سيستم master وارد سيستم slave1 مي شويم.

@ @ a@a-virtual-machine:~ a@a-virtual-machine:~\$ ssh s1@192.168.100.101 Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.13.0-36-generic 1686) * Documentation: https://help.ubuntu.com * Management: https://landscape.canonical.com * Support: https://lubuntu.com/advantage 322 packages can be updated. 158 updates are security updates. Last login: Fri Sep 14 01:54:31 2018 from 192.168.100.100 s1@s1-virtual-machine:~\$ exit logout Connection to 192.168.100.101 closed. a@a-virtual-machine:~\$

همانطور که در شکل بالا دیدید با وارد کردن دستور ssh s1@192.168.100.101 وارد سیستم s1 شدیم:

فرمت دستور به این صورت است که در ابتدا دستور ssh و در ادامه آن username و سپس @ و بعد از آن آدرس آیپی آن می آید که بعد از زدن دکمه enter با درخواست رمز مواجه می شویم. رمز را وارد میکنیم. وارد سیستم s1 می شویم. حال ما درون سیستم s1 هستیم برای اینکه بتوانیم به سیستم اصلی خود بازگردیم می توان از دستور exit استفاده کرد.

در ادامه کار نیاز داشتیم که master بدون وارد کردن رمز آن سیستم به آن سیستم وارد شود لذا کاری که کردیم به قرار زیر بود:

از آنجایی که مجموعه ابزار های استاندارد OpenSSH شامل یک ابزار به نام ssh-keygen می باشد که برای تولید کلید های جفت (key pairs) بکار می رود، لذا

کاری که کردیم این بود که یک کلید را در master تولید کردیم و آن را در تمامی slave ها کپی کردیم

برای تولید key (کلید) در Master :

در سیستم master از دستور: ssh-keygen –t rsa استفاده کردیم:



دستور ssh-keygen – trsa را وارد کردیم و سیستم از ما می پرسد که کلید را در کدام فایل ذخیره کند .با فشردن دکمه ی enter همان مکان پیش فرضی را که سیستم به ما می گوید را انتخاب می کنیم سپس همانطور که در تصویر میبینید و از آنجایی که من قبلا هم این کلید را تولید کرده بودم سیستم سوال می کند که قبلا کلید وجود دارد آیا دوباره این کار را انجام بدهد یا خیر با زدن دکمه ی ۷ قبول می کنیم که دوباره کلید را تولید کند. سپس سیستم از ما یک عبارت عبور می خواهد، با زدن دکمه ی ۷ قبول می کنیم که دوباره خالی در نظر میگیریم و هم به مرحله بعد می رویم. مرحله بعد دوباره می خواهد که عبارت عبور را وارد کنیم. دوباره enter را می زنیم. عملیات تولید زوج کلید به اتمام رسید . حال برای حصول اطمینان با زدن دستور معنی مسیر در آنجا با دستور ای می و ایت کار که انتخاب کردیم کلید ها در آنجا ذخیره شوند، می شویم. یعنی مسیر . دوباره ssh با دستور ای می در این می در این کار یا در این ای در ای در ای در ای در ای در دستور . دوباره در آنجا با دستور ای می در این کار دا در آنجا ایجاد شده اند یا خیره شوند، می شویم. یعنی مسیر

a@a-virtual-machine:~\$ ssh-keygen -t rsa Generating public/private rsa key pair. Enter file in which to save the key (/home/a/.ssh/id rsa): /home/a/.ssh/id_rsa already exists. Overwrite (y/n)? y Enter passphrase (empty for no passphrase): Enter same passphrase again: Your identification has been saved in /home/a/.ssh/id_rsa. Your public key has been saved in /home/a/.ssh/id_rsa.pub. The key fingerprint is: SHA256:840exhWk002pgPhlnUDU/AwD3ofhJbTLUk1MgsnrCrM a@a-virtual-machine The key's randomart image is: ---[RSA 2048]----+ 0+*=++ ..o*Xo . ..oBOo+ ..+=00* . . o=.o.S 0. +. ...0 .0. . .0 --[SHA256]a@a-virtual-machine:~\$ cd .ssh a@a-virtual-machine:~/.ssh\$ ls id_rsa id_rsa.pub known_hosts a@a-virtual-machine:~/.ssh\$

همانطور که در تصویر هم می بینیم، دو کلید با نام های id_rsa و id_rsa.pub تشکیل شده است.

کلید id_rsa.pub کلیدی است که باید به سیستم های slave کپی شود و آنها هم یک نسخه از آن را داشته باشند.

پس در سیستم master در دایر کتوری ssh. دستور زیر را وارد می کنیم:

scp id_rsa.pub <u>username@IP_address:~/.ssh/id_rsa.pub</u>



بعد از اینکه دستور را وارد کرده و دکمه ی enter را می زنیم با درخواست رمز عبور سیستم مقصد مواجه می شویم. رمز صحیح را وارد کرده و دکمه enter را فشار می دهیم. عملیات کپی کردن در کسری از ثانیه انجام می شویم. رمز صحیح را وارد کرده و دکمه enter را فشار می دهیم. عملیات کپی کردن در کسری از ثانیه انجام می شود. در مرحله بعد کاری که باید بکنیم این است که فایل id_rsa.pub را که به slave ها کپی کردیم را در خود سیستم های slave می slave این است که فایل authorized کپی کردن در کسری از ثانیه انجام در خود سیستم مقصد مواجه می شود. در مرحله بعد کاری که باید بکنیم این است که فایل id_rsa.pub را که به slave ها کپی کردیم را در خود سیستم های slave می دود می می می در می در مرحله بعد کاری که باید بکنیم این است که فایل slave می رویم. برای اینکار به دایر کتوری shave سیستم های slave می رویم و در آنجا دستور زیر را وارد می کنیم:

cat id_rsa.pub >> authorized_keys



حالا میاییم ببینیم که آیا به هدفمان رسیده ایم یا خیر؟

هدف این بود که بدون نیاز به وارد کردن رمز عبور، master بتواند وارد slave ها بشود.

دستور زير را وارد مي كنيم:

Ssh username@IP address

```
s1@s1-virtual-machine:~
a@a-virtual-machine:~$ ssh s1@192.168.100.101
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.13.0-36-generic i686)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
322 packages can be updated.
158 updates are security updates.
Last login: Fri Sep 14 01:57:07 2018 from 192.168.100.100
s1@s1-virtual-machine:~$
```

بسیار عالی. بدون نیاز به وارد کردن رمز عبور، master وارد slave شماره ۱ شد. حال اینکار ها را برای تمامی سیستم ها انجام می دهیم و چک میکنیم که بدون نیاز به رمز عبور می توان وارد آنها شد یا خیر.

حال قبل از اینکه به ادامه کارمان بپردازیم بهتر است که سیستم و نرم افزار های موجود در آن را به آخرین نسخه اش به روز رسانی کنیم و در عین حال تعدادی نرم افزار ضروری را برای سیستم دانلود و نصب کنیم.

اینکار ها را بهتر است برای تمامی سیستم های اعم از master و slave ها انجام بدهیم بدین منظور تنظیمات شبکه را در حالت dhcp قرار داده و تنظیمات را ریستارت کرده و سیستم ها باید به اینترنت وصل باشند بعد از اتمام نیز تنظیمات رو دوباره به حالتstatic بر می گردانیم.

بدین منظور برای اینکه سیستم را آپدیت کنیم می توان از دستور زیر استفاده کرد:

Sudo apt-get update



همانطور که در تصویر زیر می بینید سیستم در حال به روز رسانی می باشد. بعد از اینکه به روز رسانی کامل شد، مواردی نیز وجود دارد که باید در سیستم دانلود و نصب شود که به قرار زیر هستند:

Sudo apt-get <u>htop</u> : این نرم افزار به نوعی نمایش دهنده وضعیت پردازنده ها است.

Sudo apt-get build-essential : این نرم افزار پیش نیازی برای نرم افزار های بعدی که قرار است دانلود کنیم می باشد.

Sudo apt-get libncurse5-dev libncursew5-dev ؛ این نرم افزار ها پیش نیازی برای نرم افزار های بعدی که قرار است دانلود کنیم می باشد.

بعد از اتمام دانلود و نصب این نرم افزار ها میریم سراغ اصل مطلب.

به منظور پیاده سازی MPI روی سیستم نیاز داریم تا یک دایرکتوری مشترک داشته باشیم. و ما اسمش رو میزاریم: mirror

به این ترتیب پیش می رویم.

ابتدا دستور زیر را در ترمینال سیستم master وارد می کنیم تا پکیج NFS روی سیستم دانلود و نصب بشود:

Sudo apt-get install nfs-kernel-server

n [1,744 B] Get:59 http://ubuntu.cs.utah.edu/ubuntu xenial-security/multiverse i386 DEP-11 M etadata [212 B] Get:60 http://ubuntu.cs.utah.edu/ubuntu xenial-security/multiverse DEP-11 64x64 Icons [29 B] Fetched 31.6 MB in 18min 29s (28.4 kB/s) Reading package lists... Done a@a-virtual-machine:-\$ sudo apt-get install nfs-kernel-server [sudo] password for a: Reading package lists... Done Building dependency tree Reading state information... Done nfs-kernel-server is already the newest version (1:1.2.8-9ubuntu12.1). 0 upgraded, 0 newly installed, 0 to remove and 323 not upgraded. a@a-virtual-machine:-\$

همانطور که می بینید بعد از وارد کردن دستور و زدن دکمه enter سیستم از ما درخواست رمز عبور می کند، رمز را وارد کرده و ادامه می دهیم. از آنجایی که nfs-kernel-server قبلا در سیستم من نصب شده لذا با این پیام که در تصویر میبینید مواجه می شوم که می گوید قبلانصب شده ولی اگر شما برای اولین بار بخواهید آن را نصب کنید مقداری طول خواهد کشید.

بعد از اتمام نصب، در سیستم master یک فولدر با نام دلخواه را ایجاد می کنیم که در اینجا من نام آن را mirror قرار دادم.

با استفاده از دستور زیر می توانید اینکار را انجام بدهید:

Sudo mkdir /mirror



همانطور که در تصویر میبینید با پیام اخطاری روبرو شدم. زیرا قبلا فولدر mirror را ساخته ام و از قبل وجود دارد.

در مرحله بعد، مالکیت این فولدر را به طوری عوض می کنم که آن به هیچ شخص یا هیچ گروهی تعلق نداشته باشد .بدین منظور از دستور زیر استفاده می کنم:

Sudo chown nobody:nogroup /mirror



در قدم بعدی:

این دایر کتوری (mirror)، را به فولدر exports اضافه می کنیم، بدین منظور از دستور زیر استفاده می کنیم:

Sudo pico /etc/exports

با وارد کردن این دستور در ترمینال، سیستم در خواست رمز عبور می کند آن را وارد کرده و پنجره ای مانند شکل زیر باز می شود:



همانطور که در تصویر بالا میبینید، در خط آخر

mirror *(rw,sync)/

را اضافه کرده ایم:

تفسير اين خط به اين صورت است كه :

/mirror : یعنی دایر کتوری mirror را به این لیست اضافه کن.

*: يعنى اينكه تمامي آي پي ها مي توانند به اين دايركتوري دسترسي داشته باشند.

(rw,sync) : یعنی تمامی آی پی ها می توانند به این دایرکتوری بنویسند(w)، از روی آن بخوانند(r)، و عملیات همگام سازی را انجام بدهند(sync).

در آخر با زدن دکمه های ترکیبی ctrl+x می توانید این فایل را ذخیره کنید و از آن خارج شوید. برای اینکه بخواهید این تغییراتی که در etc/exports/ ایجاد کردیم تاثیر گذار باشد باید از دستور زیر استفاده کنیم.

Sudo exportfs -a



حالا در ادامه:

به سیستم های دیگر (slaves)می رویم و مراحل زیر را در تک تک آنها اجرا می کنیم:

از اتصال آنها به اینترنت مطمئن می شویم

سپس NFS common را برای آنها دانلود و نصب می کنیم.

برای اینکار می توان از دستور زیر استفاده کرد:

Sudo apt-get install nfs-common

s1@s1-virtual-machine:~
s1@s1-virtual-machine:~\$ sudo pico /etc/network/interfaces
[sudo] password for s1:
s1@s1-virtual-machine:~\$ sudo service networking restart
s1@s1-virtual-machine:~\$ sudo apt-get install nfs-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
nfs-common is already the newest version (1:1.2.8-9ubuntu12.1).
0 upgraded, 0 newly installed, 0 to remove and 323 not upgraded.
s1@s1-virtual-machine:~\$

همانطور که در تصویر نیز میبینید nfs common از قبل در سیستم من نصب شده است. لذا برای نصب کردن در اولین بار، مقداری طول خواهد کشید.

بعد از اینکار در تک تک سیستم های slave یک دایرکتوری با نام mirror مشابه master ایجاد میکنیم و دستور آن به صورت زیر خواهد بود:

Sudo mkdir –p /mirror

همانطور که قبلا گفته شد این کار را برای تک تک سیستم های slave انجام می دهیم و بعد از آن به ادامه کار می پردازیم

در ادامه در سیستم master کد زیر را وارد می کنیم تا nfs kernel server شروع بکار کند.

Sudo /etc/init.d/nfs-kernel-server start



و سپس در هر یک از سیستم های slave دستور زیر را وارد می کنیم:

Sudo mount master_IP_address:/mirror/mirror

s1@s1-virtual-machine:~\$ su [sudo] password for s1: s1@s1-virtual-machine:~\$ su s1@s1-virtual-machine:~\$ su	udo udo udo	servic	/etc/ne	etwork			
[sudo] password for s1: s1@s1-virtual-machine:~\$ su s1@s1-virtual-machine:~\$ su	udo udo	servio	ce netu	Jockir			
s1@s1-virtual-machine:~\$ su s1@s1-virtual-machine:~\$ su	udo udo	servio	ce netw	Jockin			
s10s1-virtual-machine:~S su	udo	mount	sudo service networking restart				
	- I	nounc	sudo mount 192.168.100.100:/mirror /mirror				
s1@s1-virtual-machine:~\$ d1	t - h						
Filesystem Si	ize	Used	Avail	Use%	Mounted on		
udev 41	76M	0	476M	0%	/dev		
tmpfs 10	00M	6.3M	94M	7%	/run		
/dev/sda1 :	19G	3.9G	14G	22%	1		
tmpfs 49	99M	196K	499M	1%	/dev/shm		
tmpfs 5.	. OM	4.0K	5.0M	1%	/run/lock		
tmpfs 49	99M	0	499M	0%	/sys/fs/cgroup		
tmpfs 10	00M	56K	100M	1%	/run/user/1000		
/dev/sr0 1.	.6G	1.6G	0	100%	/media/s1/Ubuntu 16.04.4 LTS i386		
192.168.100.100:/mirror	19G	4.2G	14G	24%	/mirror		
s1@s1-virtual-machine:~S							
e a de la companya d							

در ادامه برای حصول اطمینان از اینکه فایل mirror در تک تک سیستم ها mount شده یا خیر از دستور زیر استفاده میکنیم.

Df-h

که در تصویر بالا نیز می بینید که فایل 192.168.100.100:/mirror در این سیستم mount شده است. اکنون برای اینکه به درک دقیقی از این فولدر به اشتراک گذاشته شده برسیم بیاییم در سیستم master در دایرکتوری /mirror یک فایل ایجاد کنیم و سپس ببینیم که آیا این فایل ایجاد شده، در سیستم slave نیز قابل دسترسی است یا خیر:

در سیستم master در دایر کتوری /mirror یک فایل با نام دلخواه ایجاد می کنیم برای این کار از دستور زیر استفاده می کنیم:

Sudo touch test

با این دستور ما یک فایل به نام test را ایجاد کردیم. اکنون با دستور ls میبینیم که آیا test در آن دایرکتوری وجود دارد یا خیر.



اکنون در سیستم های slave :

بررسی می کنیم که آیا در دایرکتوری آنها نیز این فایل test موجود است یا خیر.



بله. همانطور که میبینید این فایل در دایرکتوری /mirror سیستم های slave ایجاد شده است.

اکنون روند کار ایجاد فولدر مشترک به اتمام رسیده است. در ادامه به نصب MPICH2 خواهیم پرداخت.

در ادامه:

در جلسه های بعدی به بررسی و نصب MPI پرداختیم. برای ادامه کارمان با یک نسخه ی بخصوص از MPI به نام MPICH2 کار کردیم و آ[«] را در سیستم master دانلود و نصب کردیم.

حال به بررسی این روند می پردازیم.

ابتدا از شبکه بودن سیستم ها اطمینان حاصل کردیم، لذا دستور زیر را در تک تک سیستم ها بررسی کردیم که آیا پکت ها ارسال می شوند یا خیر:

Ping 192.168.1.10

Ping 192.168.1.11

Ping 192.168.1.12

Ping 192.168.1.13

Ping 192.168.1.14

سپس از قابلیت ورود سیستم master به سیستم های دیگر بدون نیاز به رمز عبور با استفاده از دستور زیر مطمئن می شویم:

Ssh username@IP_Address

بعد از این دو وارد فاز اصلی کار یعنی نصب MPICH2 می شویم.

تنظیمات شبکه را به dhcp تبدیل می کنیم،

سپس MPICH2 را در دایر کتوری /mirror سیستم master دانلود می کنیم:

در سیستم مستر با وارد کردن فرمان زیر در دایرکتوری /mirror:

Sudo wget http://www.mpich.org/static/downloads/3.0.4/mpich-3.0.4.tar.gz

با مرحله دانلود mpich روبرو می شویم.



همونطور که در تصویر هم میبینید بعد از وارد کردن دستور و زدن دکمه Enter سیستم از درخواست رمز عبور می کند با وارد کردن رمز صحیح با روال دانلود روبرو می شویم.

بعد از اتمام دانلود، با زدن دستور ls محتویات دایر کتوری /mirror را چک می کنیم.



خب. همونطور که می بینید mpich-3.0.4.tar.gz به رنگ قرمز در تصویر لیست شده. کاری که در مرحله بعد باید انجام دهیم این است که فایل فشرده mpich-3.0.4.tar.gz که دانلود کردیم را Extract کنیم:

بدین منظور از دستور زیر بهره میبریم:

Sudo tar xvf mpich-3.0.4.tar.gz

🔞 🕒 🗉 a@a-virtual-machine: /mirror
mpich2/ mpich-3.0.4/ mpich-3.0.4.tar.gz
a@a-virtual-machine:/mirror\$ sudo tar xvf mpich-3.0.4.tar.gz
mpich-3.0.4/
mpich-3.0.4/www/
mpich-3.0.4/www/www1/
mpich-3.0.4/www/www1/mpicc.html
mpich-3.0.4/www/www1/mpif77.html
mpich-3.0.4/www/www1/mpicxx.html
mpich-3.0.4/www/www1/mpif90.html
mpich-3.0.4/www/www1/mpiexec.html
mpich-3.0.4/www/www3/
mpich-3.0.4/www/www3/MPI_Attr_delete.html
mpich-3.0.4/www/www3/MPI_Attr_get.html
mpich-3.0.4/www/www3/MPI_Attr_put.html
mpich-3.0.4/www/www3/MPI_Comm_create_keyval.html
mpich-3.0.4/www/www3/MPI_Comm_delete_attr.html
mpich-3.0.4/www/www3/MPI_Comm_free_keyval.html
mpich-3.0.4/www/www3/MPI_Comm_get_attr.html
mpich-3.0.4/www/www3/MPI_Comm_set_attr.html
mpich-3.0.4/www/www3/MPI_Keyval_create.html
mpich-3.0.4/www/www3/MPI_Keyval_free.html
mpich-3.0.4/www/www3/MPI_Type_create_keyval.html
mpich-3.0.4/www/www3/MPI_Type_delete_attr.html
mpich-3.0.4/www/www3/MPI_Type_free_keyval.html

بعد از اینکه عمل Extract به اتمام رسید، وارد پوشه mpich-3.0.4 می شویم و با دستور ls محتویات داخل

پوشه را بصورت لیست می بینیم:

😣 🗐 🗊 a@a-virl	tual-machine: /mirror/	mpich-3.0.4	
mpich-3.0.4/REI mpich-3.0.4/REI mpich-3.0.4/ReI mpich-3.0.4/COI mpich-3.0.4/COI mpich-3.0.4/COI mpich-3.0.4/COI mpich-3.0.4/COI mpich-3.0.4/.00 a@a-virtual-mage a@a-virtual-mage	LEASE_NOTES ADME.winbin.rtf kefile.am PYRIGHT.rtf PYRIGHT ANGES ailmap itignore chine:/mirror\$ ls 3.0.4 mpich-3.0.4 chine:/mirror\$ cd	4.tar.gz test mpich-3.0.4	
a@a-virtual-ma	chine:/mirror/mpic	ch-3.0.4\$ ls	
aclocal.m4	COPYRIGHT	man	README.envvar
CHANGES	doc	mpich-doxygen	README.WINDIN.FUT
confdb	examples	mpich-doxygen in	src
config.log	lib	mpich.sln	subsys include.m4
config.lt	libtool	mpichs.vcproj	test
config.status	maint	mpich.vcproj	winconfigure.wsf
config.system	Makefile	mpi.def	WWW
configure	Makefile.am	mpi.vcproj	
configure.ac	Makefile.in	mpl.vcproj	
contrib	makewindist.bat	README	
a@a-virtual-ma	chine:/mirror/mpic	ch-3.0.4\$	

سپس در دایرکتوری /mirror/mpich-3.0.4 دستوریا به نوعی دستورات زیر را وارد می کنیم:

Sudo ./configure -prefix=/mirror/mpich2 -disable-f77 -disable-fc

با زدن دکمه Enter وارد مراحل انجام دستورات می شویم که وقت متوسط زیادی را از شما میگیرد.

😣 🖱 🗊 a@a-virt	ual-machine: /mirror/	mpich-3.0.4	
mpich-3.0.4/REA mpich-3.0.4/Mak mpich-3.0.4/COP mpich-3.0.4/COP mpich-3.0.4/COP mpich-3.0.4/CHA mpich-3.0.4/.ma mpich-3.0.4/.gi a@a-virtual-mac mpich2 mpich-3 a@a-virtual-mac	DME.winbin.rtf efile.am YRIGHT.rtf YRIGHT NGES ilmap tignore hine:/mirror\$ ls .0.4 mpich-3.0.4	.tar.gz test	
a@a-virtual-mac aclocal.m4 autogen.sh CHANGES confdb config.log config.lt config.status config.system configure configure.ac contrib a@a-virtual-mac h2disable-f7	hine:/mirror/mpic COPYRIGHT COPYRIGHT.rtf doc examples lib libtool maint Makefile Makefile.am Makefile.in makewindist.bat hine:/mirror/mpic 7disable-fc	<pre>h-3.0.4\$ ls man mpich.def mpich-doxygen mpich-doxygen.in mpich.sln mpichs.vcproj mpi.def mpi.vcproj mpl.vcproj README h-3.0.4\$ sudo ./co</pre>	README.envvar README.winbin.rtf RELEASE_NOTES src subsys_include.m4 test winconfigure.wsf www

همانطور که در تصویر میبینید در خط آخر آن سه دستور در یک خط وارد شده که در زیر به تفسیر این دستور ها خواهیم پرداخت:

Sudo./configure <u>-prefix=/mirror/mpich2</u> <u>-disable-f77</u> <u>-disable-fc</u> ۱ ۲ ۳ دستور شماره ۱ : کار پیکربندی mpich2 را انجام میدهد. دستور شماره ۲ : غیر فعال کردن فورترن ۷۷ دستور شماره ۳ :غیر فعال کردن کامپایلر فورترن.

دستور زير را وارد مي كنيم:

Sudo make



همانطور که در تصویر میبینید سیستم درخواست رمز عبور می کند، با وارد کردن رمز عبور صحیح وارد مرحله نصب و بررسی می شویم که مقداری زمان گیر است.

بعد از اتمام پروسه sudo make دستور زیر را وارد می کنیم.

Sudo make install

800	a@a-virtual-machine: /mirror/mpich-3.0.4
CC	<pre>src/mpi_t/pvar_handle_free.lo</pre>
CC	src/mpi_t/pvar_read.lo
CC	<pre>src/mpi_t/pvar_readreset.lo</pre>
CC	src/mpi_t/pvar_reset.lo
CC	<pre>src/mpi_t/pvar_session_create.lo</pre>
cc	<pre>src/mpi_t/pvar_session_free.lo</pre>
cc	<pre>src/mpi_t/pvar_start.lo</pre>
cc	<pre>src/mpi_t/pvar_stop.lo</pre>
cc	src/mpi_t/pvar_write.lo
GEN	lib/libmpich.la
CXX	src/binding/cxx/initcxx.lo
CXXLD	lib/libmpichcxx.la
CC	src/env/mpichversion.o
CCLD	src/env/mpichversion
cp -p sr	c/env/mpicc.bash_src/env/mpicc
cp -p sr	c/env/mpicxx.bash src/env/mpicxx
make[2]:	Leaving directory '/mirror/mpich-3.0.4'
Making a	ll in examples
make[2]:	Entering directory '/mirror/mpich-3.0.4/examples'
CC	cpi.o
CCLD	cpi
make[2]:	Leaving directory '/mirror/mpich-3.0.4/examples'
make[1]:	Leaving directory '/mirror/mpich-3.0.4'
a@a-virt	ual-machine:/mirror/mpich-3.0.4\$ sudo make install

این پروسه نیز کمی زمان بر است.

بعد از اتمام پروسه

در همان دایر کتوری /mirror/mpich-3.0.4 دستور های زیر را وارد میکنیم.

Export PATH=/mirror/mpich2/bin:\$PATH

Export LD_LIBRARY_PATH="/mirror/mpich2/lib:\$LD_LIBRARY_PATH"

```
a@a-virtual-machine: /mirror/mpich-3.0.4 🗊 🖉
mpd_pmi.pdf /mirror/mpich2/share/doc/mpich/smpd_pmi.pdf ; fi
if [ -s ./doc/logging/logging.pdf ] ; then /usr/bin/install -c -m 644 ./doc/logg
ing/logging.pdf /mirror/mpich2/share/doc/mpich/logging.pdf ; fi
  mkdir -p '/mirror/mpich2/include'
 mkdir -p '/mirror/mpich2/include'
  /usr/bin/install -c -m 644 src/binding/cxx/mpicxx.h src/include/mpi.h '/mirror/
mpich2/include'
 mkdir -p '/mirror/mpich2/lib/pkgconfig'
  /usr/bin/install -c -m 644 src/packaging/pkgconfig/mpich.pc '/mirror/mpich2/lib
/pkgconfig'
make[3]: Leaving directory '/mirror/mpich-3.0.4'
make[2]: Leaving directory '/mirror/mpich-3.0.4'
Making install in examples
Making install in examples
make[2]: Entering directory '/mirror/mpich-3.0.4/examples'
make[3]: Entering directory '/mirror/mpich-3.0.4/examples'
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/mirror/mpich-3.0.4/examples'
make[2]: Leaving directory '/mirror/mpich-3.0.4/examples'
make[1]: Leaving directory '/mirror/mpich-3.0.4/examples'
a@a-virtual-machine:/mirror/mpich-3.0.4$ export PATH=/mirror/mpich2/bin:$PATH
a@a-virtual-machine:/mirror/mpich-3.0.4$ export LD_LIBRARY_PATH="/mirror/mpich2/
lib:$LD_LIBRARY_PATH"
a@a-viriual-machine:/mirror/mpich-3.0.45
```

حال در ادامه :

دستور زير را وارد ميكنيم:

Sudo pico /etc/environment

با در خواست رمز عبور مواجه خواهیم شد. رمز صحیح را وارد می کنیم



با زدن دکمه Enter پنجره ای مانند شکل زیر باز خواهد شد:

890	a@a-virtual-machine: /mirror
File	ironment is being edited (by root with nano 2.5.3, PID 18292); continue.
N No	<u>^C</u> Cancel

سیستم میگوید که فایل environment قرار تغییر کنه ، ادامه بدیم یا نه؟

با زدن دکمه ۷ کار را ادامه میدهیم.



پنجره ای مانند شکل بالا باز خواهد شد. کاری که می کنیم به این قرار است که متنی که در تصویر های لایت شده را به متن اولیه درون فایل environment اضافه میکنیم.

با زدن دکمه های ترکیبی ctrl+x و در ادامه y و سپس Enter فایل environment را ذخیره کرده و از آن خارج می شویم.

و در ادامه نیاز به htop داریم که پرو سس هایی که بطور موازی در MPI اجرا می شوند را نظارت کنیم.

در صورتیکه نرم افزار htop روی سیستم تان نصب نباشد می توانید از دستور زیر برای دانلود و نصب آن استفاده کنید:

Sudo apt-get install htop

😣 🗐 🗊 🛛 a@a-virtual-machine: ~ Hit:4 http://ubuntu.cs.utah.edu/ubuntu xenial-security InRelease Reading package lists... Done a@a-virtual-machine:~\$ sudo apt-get htop [sudo] password for a: E: Invalid operation htop a@a-virtual-machine:~\$ sudo apt-get install htop Reading package lists... Done Building dependency tree Reading state information... Done The following NEW packages will be installed: htop 0 upgraded, 1 newly installed, 0 to remove and 184 not upgraded. Need to get 81.1 kB of archives. After this operation, 213 kB of additional disk space will be used. Get:1 http://ubuntu.cs.utah.edu/ubuntu xenial-updates/universe i386 htop i386 2. 0.1-1ubuntu1 [81.1 kB] Fetched 81.1 kB in 1s (60.9 kB/s) Selecting previously unselected package htop. (Reading database ... 176645 files and directories currently installed.) Preparing to unpack .../htop_2.0.1-1ubuntu1_i386.deb ... Unpacking htop (2.0.1-1ubuntu1) ... Processing triggers for gnome-menus (3.13.3-6ubuntu3.1) ... Processing triggers for desktop-file-utils (0.22-1ubuntu5.1) ...

برای بالا آوردن نرم افزار htop در ترمینال از دستور ساده زیر استفاده می کنیم.

Htop

وقتی اینکار را کردید نرم افزار htop بالا می آید که در تصویر زیر آن را می بینید:

😣 🖨 🗈 a@a-virtual-machine: ~											
CPU[1.3%] Tasks: 104, 270 thr; 1 running Mem[
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
28058	а	20	0	5800	3316	2812	R	0.7	0.3	0:00.53	htop
1143	root	20	Θ	182M	26128	10020	S	0.0	2.6	0:27.07	/usr/lib/xorg/Xor
26944	а	20	0	858M	138M	65252	S	0.0	13.9	0:57.91	/usr/lib/firefox/
1372	а	20	0	163M	11484	8584	S	0.0	1.1	0:00.90	/usr/lib/i386-lin
27051	а	20	0	598M	104M	58744	S	0.0	10.4	0:06.54	/usr/lib/firefox/
1180	root	20	0	182M	26128	10020	S	0.0	2.6	0:01.84	/usr/lib/xorg/Xor
2020	а	20	0	116M	14296	9212	S	0.0	1.4	0:10.67	/usr/lib/gnome-te
26956	а	20	0	858M	138M	65252	s	0.0	13.9	0:00.65	/usr/lib/firefox/
27057	а	20	0	598M	104M	58744	S	0.0	10.4	0:00.25	/usr/lib/firefox/
1466	а	20	0	262M	35568	12176	S	0.0	3.5	0:25.92	compiz
1	root	20	0	24084	2976	1932	s	0.0	0.3	0:05.00	/sbin/init splash
320	root	20	0	5872	2428	2216	S	0.0	0.2	0:00.38	/lib/systemd/syst
346	root	20	0	15472	2068	2012	s	0.0	0.2	0:01.80	/lib/systemd/syst
433	systemd-	t 20	0	12616	2052	2040	S	0.0	0.2	0:00.00	/lib/systemd/syst
431	systemd-	t 20	0	12616	2052	2040	S	0.0	0.2	0:00.05	/lib/systemd/syst
449	root	20	0	3016	0	0	S	0.0	0.0	0:00.00	/usr/sbin/rpc.idm
794	root	20	0	7128	2420	2364	S	0.0	0.2	0:00.01	/usr/sbin/cron -f

در قسمت بالا سمت چپ عکس، همانطور که میبینید فقط یک cpu برای این سیستم نمایش داده می شود یعنی این سیستم دارای یک هسته است.

با فشار دادن دکمه Q می توانید از پنجره htop خارج شوید.

بعد از این کار:

در سیستم master فایل hosts در قسمت etc/ را ویرایش می کنیم.

دستور زير را وارد مي كنيم:

Sudo pico hosts

در پنجره باز شده طبق شکل زیر آی پی هر سیستم را می نویسیم.

😣 🖨 🗊 a@a-virtual-machine	e: ~
GNU nano 2.5.3	File: hosts
192.168.100.100 192.168.100.101 192.168.100.102	
<mark>^G</mark> Get Help <mark>^O</mark> Write Ou △X Fxit <u>△R</u> Read Fil	[Read 3 lines] It <mark>^W</mark> Where Is <mark>^K</mark> Cut Text <mark>^J</mark> Justify <mark>^C</mark> Cur Pos e <u>^\</u> Replace^W Uncut Text <u>^T</u> To Spell <u>^Go To Line</u>

تغييرات را ذخيره كرده و از اين پنجره خارج مي شويم.

توجه داشته باشیم که آی پی هایی که در آزمایشگاه در فایل hosts ذخیره شده اند طبق جدول زیر می باشند:

آی پی	تعداد پردازنده:
192.168.1.10	2
192.168.1.11	2
192.168.1.12	2
192.168.1.13	4
192.168.1.14	4

حال برای اجرای برنامه ها می توان به منوال زیر ادامه دهید:

اکنون که همه چیز آماده استمی توانیم نسبت به اجرای برنامه های موازی خود اقدام کنیم. بدین منظور ابتدا فایل برنامه خود را در پوشه ی mirror کپی کرده و آن را با استفاده از کامپایلر MPI متناسب با زبان برنامه، کامپایل می کنیم. بعنوان مثال برای کامپایل کردن یک برنامه زبان C از دستور زیر کمک می گیریم:

/nfsshare\$ mpicc -o mpi_sample mpi_sample.c

حال می توان به اجرای برنامه کامپایل شده توسط MPICH پرداخت.

بدین منظ.ر به اجرای دستور زیر می پردازیم:

/nfsshare\$ mpirun -f hosts -n 16 ./mpi_sample

بجای عدد ۱۶ می توان تعداد پردازنده هایی که قصد دارید توسط برنامه در گیر شوند را وارد نمود. همچنین بجای عبارت mpi_sample نیز نام برنامه های خود را وارد خواهیم کرد.

پس از این کار برنامه به خوبی بر روی تمام سیستم ها اجرا گشته و می توان به کمک htop نسبت به مشاهده عملکرد پردازنده ها بر روی هر کدام از کامپیوتر ها اقدام کرد.

- ۱. سایت دانشگاه گیلان guilan.ac.ir
- /http://mpitutorial.com/tutorials/running-an-mpi-cluster-within-a-lan .٢