

« به نام او »

آموزش شبیه سازی دو بعدی فوتبال

نویسندگان : محمدعلی میرزایی - علی یعقوبی

مرجع روبوکاپ ایران

www.iranrcss.com

مباحث این جلسه :

- مروری بر تاریخچه لینوکس و آشنایی با توزیع ها و خط فرمان
- سیستم شبیه سازی دو بعدی فوتبال

گنو لینوکس



گنو/لینوکس جدا از فلسفه و تفکر آزادی که پشت آن است، سیستم عاملی است بر اساس هسته لینوکس که شامل ابزارهای GNU، برنامه نصاب، سیستم مدیریت بسته ها و بسیاری از نرم افزارهای مورد نیاز شما باشد. برخلاف سیستم عامل ویندوز مایکروسافت که توسط یک کمپانی تولید و پشتیبانی می شود، گنو/لینوکس را افراد متخصص مختلف و شرکت های بسیاری تولید، توزیع و پشتیبانی می کنند. به علت اینکه تمامی اجزای گنو/لینوکس آزاد و رایگان هستند، همه می توانند بر اساس نیاز خود یک نسخه سفارشی از آن تولید کنند. به همین خاطر توزیع های گنو/لینوکس از تنوع بسیار زیادی برخوردارند و قدرت انتخاب مناسبی به کاربران خود می دهند که در برخی مواقع این تنوع باعث سردرگمی کاربران در انتخاب توزیع مناسب می شود. از سال ۱۹۹۳ به بعد، افراد و شرکت های مختلفی توزیع های سفارشی شده خود را وارد بازار جهانی کرده اند که هم اکنون بیش از ۳۵۰ توزیع فعال گنو/لینوکس موجود است.



Ubuntu

یکی از توزیع های بسیار خوب و کاربردی از گنو/لینوکس به نام اوبونتو (Ubuntu) چند وقتی است چه در میان خارجی ها و چه در میان ایرانی ها محبوبیت خاصی پیدا کرده است. اما این محبوبیت در میان ایرانی ها مخصوصاً وبلاگ نویسان طوری فراگیر شده که باعث نادیده گرفتن و کم توجهی به توزیع های قدرتمند و معروف دیگر شده است.

سال ۲۰۰۴ توزیعی که تا قبل از آن سال اسم و رسمی نداشت به یکباره به یکی از محبوب ترین و معروف ترین توزیع های گنو/لینوکس تبدیل شد. اوبونتو بر اساس دبیان توسعه داده شده و شامل ویژگی هایی است که کار کردن با لینوکس را برای کاربران مبتدی و تازه کار آسان تر می کند. ویژگی هایی همچون پشتیبانی قوی، آپدیت های فراوان، نصب آسان، پکیج های نرم افزاری بسیار، مستندات کامل و رابط کاربری مناسب، اوبونتو را به کاربر پسندترین توزیع گنو/لینوکس برای کاربردهای دسکتاپ و شخصی تبدیل نموده است. اوبونتو برای استفاده های عمومی بوجود آمده است و توسعه آن به سرعت پیش می رود و هر نسخه از نسخه قبلی، کارآمدتر و قدرتمند تر می شود. درکل اوبونتو در دنیای سیستم عامل های دسکتاپ آینده ی خوبی در پیش خواهد داشت. یکی از مهمترین ویژگی های این توزیع، وجود تعداد کاربران بسیار زیاد و اجتماع های کاربری مناسب است.



Fedora

یکی از معروف ترین نسخه های گنو/لینوکس است که توسط یکی از بزرگترین شرکت های دنیای لینوکس یعنی RedHat پشتیبانی می شود. این توزیع بسیار قدرتمند دارای کاربران بسیار زیادی است و در دو نسخه دسکتاپ و سرور عرضه می شود. آپدیت های پیاپی، پشتیبانی مناسب، نصب آسان، تنظیمات پیشرفته، پکیج های نرم افزاری زیاد، فدورا را به یکی از انتخاب های اصلی کاربران گنو/لینوکس تبدیل نموده است. به علت پشتیبانی مناسب RedHat از این توزیع، شرکت های بسیار زیادی فدورا را به عنوان سیستم عامل خود انتخاب کرده اند و از آن برای کارهای مختلف خود بهره می برند. امنیت، پایداری و کارایی فدورا زبان زد بسیاری از کاربران لینوکس است



OpenSUSE

از زمانی که توزیع Suse بوجود آمد، سازندگان آن روی کاربران کامپیوترهای شخصی زوم کرده بودند. پس از اینکه در سال ۲۰۰۳ شرکت Novell شرکت تولید کننده Suse را خریداری کرد، دریافت آن را از اینترنت برای عموم میسر نمود. سپس ناول نام Suse را به OpenSUSE تغییر داد و آن را به صورت کاملاً آزاد در اختیار همه قرار داد. هم اکنون OpenSUSE انتخاب اول

کاربران حرفه ای برای استفاده های شخصی است. دسکتاپ پیش فرض آن KDE است و از برنامه نصاب بسیار قدرتمندی بهره می برد. پایداری، ظاهر چشمگیر، گرافیک قوی، امنیت، تنظیمات پیشرفته از ویژگی هایی است که کاربران OpenSUSE از آن یاد خواهند کرد .



Knoppix

این توزیع از گنو/لینوکس یک Live-CD است؛ این بدین معنی است که کاربر می تواند سیستم خود را به صورت مستقیم از روی سی دی بوت کند بدون اینکه نیاز به نصب سیستم عامل روی هارد دیسک خود داشته باشد. Knoppix قدرت بسیار زیادی در شناسایی اتوماتیک سخت افزارهای شما دارد (که حتی از دیگر توزیع های لینوکس در این زمینه قوی تر است) و شامل بسیاری از نرم افزارهای مورد نیازتان می شود. این سی دی می تواند به عنوان یک ابزار ریکاوری یا برای تست سخت افزارهای سیستم به کار برود و یا حتی به عنوان یک توزیع گنو/لینوکس مناسب برای کاربردهای دسکتاپ انتخاب شود. این توزیع محبوب بر اساس دبیان توسعه داده شده است و دارای آپدیت ها و پشتیبانی مناسبی است.



Debian

یکی از قدیمی ترین توزیع های گنو/لینوکس است که توزیع های بسیاری همچون اوبونتو بر اساس آن گسترش یافته اند. دبیان پروژه ای آزاد و غیرتجاری است که هم اکنون بیش از هزاران توسعه دهنده در سراسر دنیا بر روی نسخه های مختلف آن کار می کنند. این پروژه بسیار بزرگ است و یازده سکوی سخت افزاری را که شامل PowerPC، x۸۶، SPARC و AMD۶۴ نیز می شوند را پشتیبانی می کند. دبیان یکی از مقیدترین توزیع های گنو/لینوکس به فلسفه گنو/لینوکس و سیستم عامل یونیکس است، پایدار است و دارای یکی از بزرگترین اجتماعات کاربری لینوکس است. دبیان همچنین دارای مستندات بسیاری است و به تعداد زیادی از زبان های دنیا ترجمه شده است. فقدان ابزارهای گرافیکی و نصاب گرافیکی استفاده از دبیان را برای بسیاری از کاربران سخت نموده است. در حقیقت ماندن در دبیان در مقایسه با توزیع های دیگر مشکل تر است. بر همین اساس کاربرانی که قصد استفاده از لینوکس در کامپیوترهای شخصی را دارند بیشتر به طرف توزیع های بر اساس دبیان مانند اوبونتو کشیده می شوند. دبیان انتخاب مناسبی برای استفاده در کامپیوترهای سرویس دهنده (Server) خواهد بود.



Parsix

پارسیکس یک توزیع بسیار انعطاف پذیر، مناسب، پایدار و ایرانی برای سیستم های دسکتاپ است که به صورت دیسک زنده یا Live CD منتشر می شود اما قابلیت نصب روی هارد دیسک را نیز داراست. این توزیع برگرفته شده از توزیع KANOTIX و مبتنی بر Debian است. دسکتاپ آن به صورت پیش فرض GNOME است و نرم افزارهای پیش فرض فارسی همچون دیکشنری xFarDic و فونت های فارسی پروژه FPF در این توزیع در دسترس اند. همچنین در پارسیکس به صورت پیش فرض از فارسی نویسی پشتیبانی می شود و شما می توانید همانند ویندوز با فشردن همزمان دکمه های Alt+Shift میان زبان فارسی و انگلیسی سوئیچ کنید. پارسیکس یکی از مناسب ترین توزیع ها برای شروع یادگیری لینوکس در میان فارسی زبانان است. البته در نظر داشته باشید که پارسیکس می تواند جایگزین مناسبی برای سیستم عامل ویندوز باشد و می توانید از آن به عنوان یک سیستم عامل مستقل استفاده نمایید.

توزیع های بسیار و معروف دیگری نیز هست که میتوانید با جستجو در اینترنت، اطلاعاتی راجع به آنها بدست آورید.

خط فرمان در لینوکس :

ابزاری که در کلیه توزیع های لینوکس قرار دارد محیط ترمینال است. شما میتوانید بسیاری از کارهایتان را با استفاده از این محیط قدرتمند انجام دهید. حتی میتوانید در این محیط به جست و جو در صفحات وب پردازید!

در این بخش ما به بررسی بعضی کامند هایی که بیشتر مورد استفاده قرار میگیرد میپردازیم.

برای شروع کار ابتدا ترمینال سیستم عامل خود را باز کنید.

نکته : بعد از هر مرحله، برای پاک شدن نوشته های قبلی در ترمینال میتوانید از دکمه های **Ctrl+L** استفاده کنید.

لیست بعضی کامند ها :

- **ls**

از این دستور برای نمایش فایل های درون یک پوشه استفاده میشود. اگر این دستور را به تنهایی در ترمینال اجرا کنید فقط لیست فایل های درون فولدري که در آن قرار دارید را نشان میدهد ولی میتوانید با این دستور محتویات یک پوشه دیگر را نیز مشاهده کنید. برای مثال :

ls /Desktop

- **mkdir DIRECTORYNAME**

از این دستور برای ساخت دیرکتوری استفاده میشود. در قسمت دوم بجای **DIRECTORYNAME** باید اسم دیرکتوری مورد نظرتون را بنویسید.

- **cd**

از این دستور برای مرور فولدرها و انجام کارهایتان در آنها استفاده میکنید. برای مثال اگر در ترمینال خود بنویسید **cd Downloads/** به پوشه داندلود های گنو/لینوکس خود منتقل میشوید. این یکی از پرکاربرد ترین دستورات خط فرمان است.

- cp

از این دستور برای کپی یک فایل از یک دیرکتوری به دیرکتوری دیگر استفاده میشود. برای مثال اگر فایل **test.png**

در پوشه تصاویر لینوکس باشد و ما بخواهیم آنرا در دسکتاپ کپی کنیم باید این دستور را وارد کنیم :

cp Pictures/test.png Desktop/

- mv

استفاده از این دستور همانند دستور بالا است با این تفاوت که با این دستور، فایل کپی نمیشود بلکه بطور کامل به

پوشه مورد نظر منتقل میشود.

- rm

برای حذف فایل ها و پوشه ها (**rm -r**) استفاده میشود. بدینگونه استفاده میشود که پس از ایجاد

فاصله بعد از **rm** آدرس فایل مورد نظر را وارد میکنید.

- Locate

برای جستجوی یک فایل خاص در میان کل فایل هایتان از این دستور استفاده میشود. برای مثال :

locate test.png

- chmod

برای تغییر **permission** فایل ها و فولدر ها از این دستور استفاده میشود. برای مثال میخواهید

permission های پوشه **robocup** در پوشه **Desktop** را به ۷۷۷ تغییر دهید. باید این دستور

را وارد کنید :

chmod -R ۷۷۷ Desktop/robocup

- > filename.filetype

برای ایجاد فایل با فرمت خاص، از این دستور استفاده میشود. بعد از نوشتن کاراکتر ">" اسم فایل را نوشته، کاراکتر "." را وارد کرده و نوع فایل را مینویسید.

همچنین برای نصب بسیاری از برنامه ها باید مراحل زیر طی شود :

ابتدا به پوشه ای که فایل های برنامه در آن قرار دارد رفته و کامند های زیر را به ترتیب بزنید :

نکته : در صورت بروز ارور در هر مرحله باید ارور های مربوطه را خودتان یا با یاری گرفتن از دیگران برطرف کنید.

بیشتر این ارور ها مربوط به پکیج های مورد نیاز برای اجرای برنامه است که هنوز نصب نشده است.

- ۱) configure
- ۲) make
- ۳) make install

البته در سیستم عامل شما، واسط گرافیکی برای نصب نرم افزار ها وجود دارد که با

استفاده از آن میتوانید براحتی، پکیج های مورد نظرتان را نصب نمایید.

از این پس، وارد آموزش های شبیه سازی فوتبال دوبعدی میشویم و در صورت نیاز، آموزش های تکمیلی لینوکس را در میان مطالب، قرار میدهیم.

❖ سیستم شبیه ساز دو بعدی :

در این جا نسخه ی شبیه ساز دو بعدی ۱۵ می باشد. سیستم شبیه ساز دو بعدی شامل ۳ قسمت اصلی به

شرح زیر می باشد:

- نمایشگر فوتبال
- نمایشگر مجدد بازی
- server فوتبال

سیستم شبیه ساز دو بعدی، دو نمایشگر بازی دارد که به طور مستقیم با server ارتباط دو سویه دارد و اطلاعات تولید شده از طرف server را به صورت online یا offline نمایش می دهند که روش نمایش online در هنگام انجام یک مسابقه ی واقعی و روش نمایش offline در زمان تست و یا مطالعه روی چگونگی عملکرد یک تیم در یک بازی انجام شده، مورد استفاده قرار می گیرد. علاوه بر نمایشگرها این مجموعه شامل یک نمایشگر مجدد بازی است که اطلاعات مربوط به یک بازی انجام شده را مجدداً با اتصال به نمایشگر به تصویر می کشد و ارتباط آن با نمایشگر موجب نمایش بازی به صورت Offline می گردد.

server، نرم افزار اصلی در طراحی، ساخت و مدلسازی محیط فوتبال است. همه ی اطلاعات در ارتباط به حسگرها، عملگرها، الگوریتم های ایجاد خطا و الگوهای مورد استفاده برای ایجاد ارتباط میان عامل ها (agent) توسط سیستم شبیه ساز تولید می شود و سپس پردازش می شود و بین agent ها منتقل می گردد.

❖ شبیه‌ساز فوتبال دو بعدی :

برای اجرای برنامه‌های agent های هوشمند فوتبالیست احتیاج به یک شبیه‌ساز فوتبال داریم که در مسابقات Robocup با نام سرور فوتبال "Soccer Server" شناخته می‌شود. سرور فوتبال سیستمی است که به agent های خودمختار که برنامه‌هایی هستند که به زبان‌های مختلف نوشته شده‌اند اجازه می‌دهد در مقابل یک‌دیگر فوتبال بازی کنند. این سرویس دهنده‌ی فوتبال وظایفی شامل اعمال قوانین فیزیکی، پیش بردن زمان اجرای قوانین داور، اعمال UDP/IP دستورات agent ها، رساندن اطلاعات بازی به agent ها و ... را به عهده دارد. برای انجام بازی از معماری سرور و کلاینت استفاده شده به این صورت که سرور فوتبال به عنوان سرویس دهنده و agent ها به عنوان کلاینت که به سرور متصل می‌شوند محسوب می‌شوند. پروتکل‌های اتصال آن‌ها بسته به لیگ و ویرایش آن‌ها می‌باشد به طور مثال شبیه‌ساز فوتبال دو بعدی از UDP/IP استفاده می‌کند. ولی شبیه‌ساز سه بعدی از TCP/IP و اختلافات بسیاری مانند نحوه زمانبندی و اختصاص زمان به agent ها و ... در ویرایش‌های گوناگون آن‌ها وجود دارد .

❖ ویژگی های زمین بازی :

زمین بازی مستطیلی به ابعاد $68 * 105$ می باشد و عرض هر دروازه آن $14/02m$ است . مرکز محور مختصات بر مرکز زمین منطبق است و محور X ها از $-52/5$ تا $52/5$ و محور Y ها از -34 تا 34 می باشد . در اطراف زمین بازی ، ۵۹ پرچم در مکان های خاصی وجود دارد که به بازیکن در تعیین موقعیت خود کمک می کند .

❖ بازیکنان :

بازیکنان در محیط شبیه سازی دو بعدی ، به شکل دایره می باشند و همه ی فاصله ها و زاویه ها نسبت به مرکز داده ها اندازه گیری می شود . توپ بازی هم به شکل دایره می باشد .

❖ بازیکنان نامتشابه :

در یک تیم همه ی بازیکنان به جز دروازه بان می توانند تعویض شوند . هنگامی که بازی در مد before-kick-of قرار دارد ، server لیست تمامی بازیکنان را به همه ی مربی های online و بازیکنان می دهد . این ویژگی از نسخه ۷ به سیستم شبیه سازی فوتبال دو بعدی اضافه شده است . مربی بازیکنان دلخواه خود را به تعداد player-types که مقدار ۱۷ را دارد (نوع صفر در این مجموعه پیش فرض است) با توجه به استراتژی بازی انتخاب می کند . بازیکنان مختلف از نظر سرعت و انرژی و سایر پارامترها با هم متفاوت هستند . در جریان بازی هم مربی می تواند فقط سه بار ، یعنی به تعداد sub-max تعویض انجام دهد و با هر تعویض کلیه ی ویژگی های خصوصیاتی نظیر انرژی و تلاش به بیشترین مقادیر خود می رسد و خصوصیاتی مانند سرعت و انرژی با توجه به نوع بازیکنان جدید مقادیر مناسب را می گیرند.

❖ قوانین مسابقه :

شبه ساز دو بعدی فوتبال دارای دو سری قوانین می باشد :

۱. قوانین server : به این ترتیب است که server ، قوانینی را که از طریق محاسباتی و یا منطقی قابل تشخیص می باشند ، به شکل خودکار اعمال می کند. از قوانین محاسباتی می توان به offside و یا تنظیم زمان ۵ دقیقه ای یا ۳۰۰۰ سیکلی در هر نیمه از بازی و حفظ فاصله m ۹/۱۵ از سوی بازیکنان هر تیم در هنگام ضربه ی آزاد تیم مقابل اشاره نمود. از قوانین منطقی نیز می توان کنترل وضعیت در هریک از مدهای مختلف بازی مانند kick_in ، goal_kick و... را نام برد .

۲. قوانین انسانی : قوانینی که توسط server قابل تشخیص نیست . بنابراین داور انسانی باید آن را انجام دهد . این قوانین از نظر عملکرد منطقی به ظاعر مشکلی ندارند ، اما در روند اجرای سالم بازی را دچار اغتشاش و اختلال می کند و server قادر به تشخیص آن نمی باشد . به طور مثال احاطه کردن توپ ، پر کردن دروازه با تعداد زیادی از بازیکنان و ... که توسط داور انسانی باید اعمال شود .

❖ کار بر روی Base Source Code :

از آنجا که ما می خواهیم روی base code آماده ی فوتبال دو بعدی کار کنیم پس باید کد های base را داشته باشیم و آن ها را تغییر دهیم و سپس نتیجه ی کار را مشاهده کنیم .

برای اینکه ما از Base Source Code استفاده کنیم باید آن را به اصطلاح compile کنیم برای این منظور معمولا باید مراحل زیر را طی کنیم :

۱. باید در ابتدا درون پوشه ی Base Source Code دستور زیر را وارد کنیم :

`./configure`

۲. بعد از اتمام دستور بالا ، دستور زیر را وارد می کنیم :

`make`

۳. اگر دستورات بالا بدون خطا به اتمام برسد ، تیم آماده ی اجرا بر روی سرور شبیه ساز می باشد .

❖ **طریقه نصب سرور و مانیتور شبیه ساز دو بعدی فوتبال :**

۱. نصب سرور شبیه ساز دوبعدی فوتبال : برای این منظور مراحل زیر را به ترتیب انجام می دهیم :

✓ ابتدا وارد پوشه ی سرور شده و دستور زیر را وارد می کنیم :

```
./configure
```

```
trix@trix-inspiron-N5010: ~/Documents/RoboCup/RoboCup2D/rcsoccersim/15/rcssserver-15.0.1
trix@trix-inspiron-N5010:~/Documents/RoboCup/RoboCup2D/rcsoccersim/15/rcssserver-15.0.1$ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... [ ]
```

```

- trk@trkic-nasplon-N5910: ~/Documents/RoboCup/RoboCup2D/rcsscoursim/15/rcsscoursim-15.0.1
-
checking if g++ supports -c.o file... yes
checking if g++ supports -c.o file... (cached) yes
checking whether the g++ linker (/usr/bin/ld) supports shared libraries... yes
checking dynamic linker characteristics... GNU/Linux ld.so
checking how to hardcode library paths into programs... immediate
checking for boostlib >= 1.32.0... yes
checking whether the Boost::System library is available... yes
checking for exit in -lboost system-att... yes
checking whether the Boost::Filesystem library is available... yes
checking for exit in -lboost filesystem-att... yes
checking for select... yes
checking sys/select.h usability... yes
checking sys/select.h presence... yes
checking for sys/select.h... yes
checking for sys/socket.h... (cached) yes
checking types of arguments for select... int,fd set,*struct timeval*
configure: creating ./config.status
config.status: creating Makefile
config.status: creating rcscbase/Makefile
config.status: creating rcscbase/net/Makefile
config.status: creating rcscbase/config/Makefile
config.status: creating rcscbase/gzip/Makefile
config.status: creating src/Makefile
config.status: creating src/rcsscoursim
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing depfiles commands
config.status: executing libtool commands
trk@trkic-nasplon-N5910: ~/Documents/RoboCup/RoboCup2D/rcsscoursim/15/rcsscoursim-15.0.1

```

✓ بعد از اتمام دستور بالا ، کد زیر را وارد می کنیم :

make

```

trix@trix-Inspiron-N5010:~/Documents/RoboCup/RoboCup2D/rscsocsim/15/rcsserver-15.0.1 -
trix@trix-Inspiron-N5010:~/Documents/RoboCup/RoboCup2D/rscsocsim/15/rcsserver-15.0.15 clear

trix@trix-Inspiron-N5010:~/Documents/RoboCup/RoboCup2D/rscsocsim/15/rcsserver-15.0.15 make
make all-recursive
make[1]: Entering directory '/home/trix/Documents/RoboCup/RoboCup2D/rscsocsim/15/rcsserver-15.0.1'
Making all in rcssbase
make[2]: Entering directory '/home/trix/Documents/RoboCup/RoboCup2D/rscsocsim/15/rcsserver-15.0.1/rcsbase'
make[3]: Entering directory '/home/trix/Documents/RoboCup/RoboCup2D/rscsocsim/15/rcsserver-15.0.1/rcsbase/
/bin/bash -x /libtool -tag-CXX -mode-compile g++ -DHAVE_CONFIG_H -I. -I../ -I../ -Iusr/include -
-wall -g -O2 -MT addr.c -MD -MP -MF $deps/addr.cpo -c addr.c -o addr.cpo
libtool: compile: g++ -DHAVE_CONFIG_H -I. -I../ -I../ -Iusr/include -Wall -g -O2 -MT addr.c -MD -MP
-MF $deps/addr.cpo -c addr.cpo -FPIC -DPIC -o .libs/addr.cpo
/bin/bash -x /libtool -tag-CXX -mode-compile g++ -DHAVE_CONFIG_H -I. -I../ -I../ -Iusr/include -W
-wall -g -O2 -MT socket.c -MD -MP -MF $deps/socket.cpo -c socket.c -o socket.cpo
libtool: compile: g++ -DHAVE_CONFIG_H -I. -I../ -I../ -Iusr/include -Wall -g -O2 -MT socket.c -MD -MP
-MF $deps/socket.cpo -c socket.cpo -FPIC -DPIC -o .libs/socket.cpo
mv -f $deps/socket.cpo $deps/socket.cpo
/bin/bash -x /libtool -tag-CXX -mode-compile g++ -DHAVE_CONFIG_H -I. -I../ -I../ -Iusr/include -W
-wall -g -O2 -MT socketstream.c -MD -MP -MF $deps/socketstream.cpo -c socketstream.c -o socketstream.cpo
libtool: compile: g++ -DHAVE_CONFIG_H -I. -I../ -I../ -Iusr/include -Wall -g -O2 -MT socketstream.cpo
-MF $deps/socketstream.cpo -c socketstream.cpo -FPIC -DPIC -o .libs/socketstream.cpo
mv -f $deps/socketstream.cpo $deps/socketstream.cpo
/bin/bash -x /libtool -tag-CXX -mode-compile g++ -DHAVE_CONFIG_H -I. -I../ -I../ -Iusr/include -W
-wall -g -O2 -MT upsocket.c -MD -MP -MF $deps/upsocket.cpo -c upsocket.c -o upsocket.cpo
libtool: compile: g++ -DHAVE_CONFIG_H -I. -I../ -I../ -Iusr/include -Wall -g -O2 -MT upsocket.c -MD -MP
-MF $deps/upsocket.cpo -c upsocket.cpo -FPIC -DPIC -o .libs/upsocket.cpo
mv -f $deps/upsocket.cpo $deps/upsocket.cpo

```

[illegible]

✓ و در انتها :

sudo make install

```
trix@trix-Inspiron-N5010: ~/Documents/RoboCup/RoboCup2D/rcsoccersim/15/rcssserver-15.0.1
Libraries have been installed in:
  /usr/local/lib

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
- add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
  during execution
- add LIBDIR to the 'LD_RUN_PATH' environment variable
  during linking
- use the '-Wl,-rpath -Wl,LIBDIR' linker flag
- have your system administrator add LIBDIR to '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
.....
test -z "/usr/local/bin" || /bin/mkdir -p "/usr/local/bin"
/bin/bash ./libtool --mode=install /usr/bin/install -c rcssserver rcssclient '/usr/local/bin'
libtool: install: /usr/bin/install -c .libs/rcssserver /usr/local/bin/rcssserver
libtool: install: /usr/bin/install -c .libs/rcssclient /usr/local/bin/rcssclient
test -z "/usr/local/bin" || /bin/mkdir -p "/usr/local/bin"
/usr/bin/install -c rcsoccersim '/usr/local/bin'
test -z "/usr/local/include/rcssserver" || /bin/mkdir -p "/usr/local/include/rcssserver"
/usr/bin/install -c *.h 644 clangparser.h coach lang tok.h clangbuilder.h clangmsgbuilder.h clangmsg.h clangmet
msg.h clangfreeformmsg.h clangunusupmsg.h clangrules.h clangdelmsg.h clanginfo.h clangadvicemsg.h clangde
fmsg.h clangaction.h clangutil.h coach lang comp.h coach lang parser.h arithop.h compop.h cond.h region.h rule.
h vector.h rcssexceptions.h '/usr/local/include/rcssserver'
```

```
trix@trix-Inspiron-N5010: ~/Documents/RoboCup/RoboCup2D/rcsoccersim/15/rcssserver-15.0.1
during linking
- use the '-Wl,-rpath -Wl,LIBDIR' linker flag
- have your system administrator add LIBDIR to '/etc/ld.so.conf'

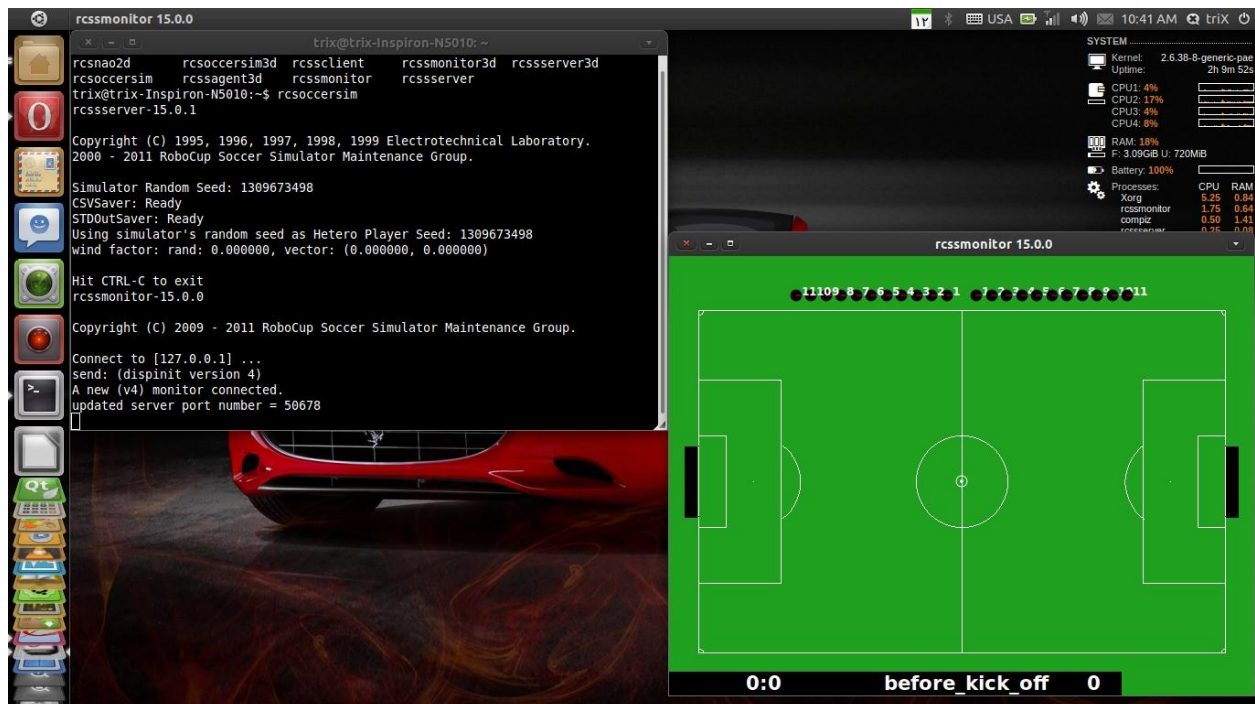
See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
.....
test -z "/usr/local/bin" || /bin/mkdir -p "/usr/local/bin"
/bin/bash ./libtool --mode=install /usr/bin/install -c rcssserver rcssclient '/usr/local/bin'
libtool: install: /usr/bin/install -c .libs/rcssserver /usr/local/bin/rcssserver
libtool: install: /usr/bin/install -c .libs/rcssclient /usr/local/bin/rcssclient
test -z "/usr/local/bin" || /bin/mkdir -p "/usr/local/bin"
/usr/bin/install -c rcsoccersim '/usr/local/bin'
test -z "/usr/local/include/rcssserver" || /bin/mkdir -p "/usr/local/include/rcssserver"
/usr/bin/install -c *.h 644 clangparser.h coach lang tok.h clangbuilder.h clangmsgbuilder.h clangmsg.h clangmet
msg.h clangfreeformmsg.h clangunusupmsg.h clangrules.h clangdelmsg.h clanginfo.h clangadvicemsg.h clangde
fmsg.h clangaction.h clangutil.h coach lang comp.h coach lang parser.h arithop.h compop.h cond.h region.h rule.
h vector.h rcssexceptions.h '/usr/local/include/rcssserver'
make[3]: Leaving directory '/home/trix/Documents/RoboCup/RoboCup2D/rcsoccersim/15/rcssserver-15.0.1/src'
make[2]: Leaving directory '/home/trix/Documents/RoboCup/RoboCup2D/rcsoccersim/15/rcssserver-15.0.1/src'
make[1]: Leaving directory '/home/trix/Documents/RoboCup/RoboCup2D/rcsoccersim/15/rcssserver-15.0.1/src'
Making install in .
make[1]: Entering directory '/home/trix/Documents/RoboCup/RoboCup2D/rcsoccersim/15/rcssserver-15.0.1'
make[2]: Entering directory '/home/trix/Documents/RoboCup/RoboCup2D/rcsoccersim/15/rcssserver-15.0.1'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Leaving directory '/home/trix/Documents/RoboCup/RoboCup2D/rcsoccersim/15/rcssserver-15.0.1'
make[1]: Leaving directory '/home/trix/Documents/RoboCup/RoboCup2D/rcsoccersim/15/rcssserver-15.0.1'
trix@trix-Inspiron-N5010: ~/Documents/RoboCup/RoboCup2D/rcsoccersim/15/rcssserver-15.0.1
```

۲. نصب مانیتور شبیه ساز دوبعدی فوتبال : دستورات بالا را عینا تکرار می کنیم .

❖ اجرای محیط شبیه ساز دو بعدی فوتبال :

دستور زیر را در ترمینال وارد می کنیم :

rcsoccersim



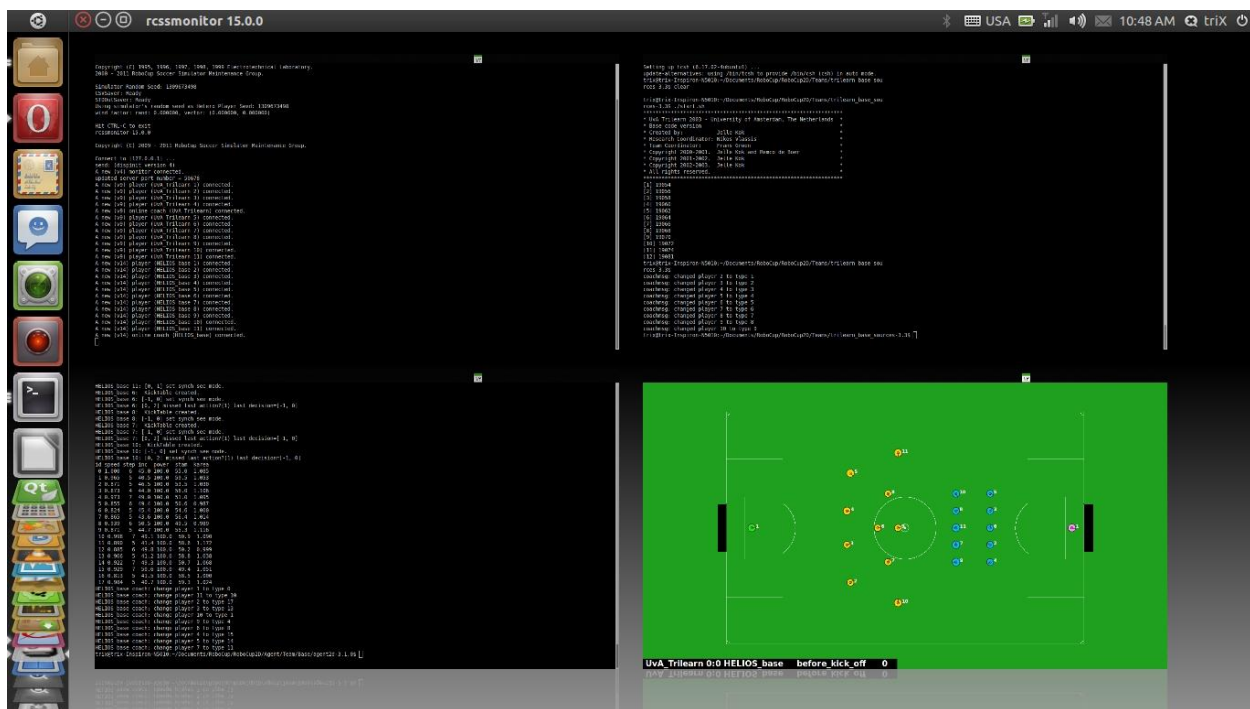
❖ اجرا کردن تیم ها :

برای اینکار باید از Bash فایللی که درون هر تیم وجود دارد استفاده کنیم . بعد از کامپایل کردن هر

یک از تیم ها وارد پوشه ی تیم مورد نظر رفته و دستور زیر را وارد می کنیم :

./BashFileName

البته به جای BashFileName نام اسکریپت اجرا کننده تیم را وارد می کنیم .



منابع :

- سایت شخصی محمدعلی میرزایی mamirzaee.com

- وب سایت اوبونتوی فارسی Ubuntu.ir

پایان جلسه دوم