

Narration	Time
رڈارکیشن اور پائپس کے بارے میں اس سپوکن ٹوریل میں آپ کا استقبال ہے۔	00:00
میں ابٹو 10.04 استعمال کر رہا ہوں۔	00:07
ہم یہ مانتے ہیں کہ آپ کو پہلے سے ہی لینکس آپریٹنگ سسٹم اور کمانڈس کے بارے میں کچھ بنیادی معلومات ہیں۔	00:09
اگر آپ چاہیں تو درج ذیل ویب سائٹ پر یہ دوسرے سپوکن ٹوریلز کے ذریعے دستیاب ہیں۔	00:16
یاد رہے کہ لینکس کیس سینسیٹو ہے۔	00:22
اس ٹوریل میں استعمال تمام کمانڈس lower case حروف میں ہیں، دوسری صورت مبینہ ہیں۔	00:25
لینکس میں زیادہ تر کام ہم ٹرمینل کے ذریعے کرتے ہیں۔	00:32
جب ہم کمانڈ چلاتے یعنی execute کرتے ہیں، ہم عام کی بورڈ سے ٹائپ کرتے ہیں۔	00:35
مانیں کہ ہمیں وقت اور تاریخ پتہ کرنا ہے	00:39
ہم صرف کی بورڈ سے "ڈیٹ" ٹائپ کرتے ہیں اور اینٹر دباتے ہیں	00:41
لہذا ہم عام کی بورڈ سے ان پٹ دیتے ہیں۔	00:46
اسی طرح ہم دیکھتے ہیں کہ ہماری کمانڈ کا آؤٹ پٹ ٹرمینل ونڈو پر بھی ظاہر ہوتا ہے۔	00:48
یہ بھی مانیں کہ جب ہم کسی کمانڈ کو execute کرتے ہیں کچھ ایرر بھی آتے ہیں	00:56
مثلاً ہم "cat space aaa" ٹائپ کرتے ہیں اور اینٹر دباتے ہیں۔	00:59
اب ایسے نام سے کوئی بھی فائل موجود نہیں ہے	01:05
لہذا ایک ایرر ظاہر ہوتا ہے	01:08
اب یہ ایرر ٹرمینل ونڈو پر بھی آتا ہے، اس لئے ہم ایرر رپورٹنگ ٹرمینل پر بھی دیکھتے ہیں۔	01:10
اب اینپگ، آؤٹ پٹنگ اور ایرر رپورٹنگ کمانڈس سے متعلق تین خاص کام ہیں	01:20
رڈارکیشن کے بارے میں سیکھنے سے پہلے ہمیں دو اہم اصولوں کے بارے میں جاننا چاہئے۔ جو ہے سٹریم اور فائل ڈسکرپٹر۔	01:24
لینکس شیل جیسے کے بیش، ان پٹ کو قبول کرتا ہے اور ترتیب کے طور پر حروف کی سٹریم میں آؤٹ پٹ بھیجتا ہے۔	01:31
ہر حرف اس کے پہلے یا اس کے بعد والے سے آزاد ہوتا ہے۔	01:37
سٹریمس تک فائل IO تکنیک کا استعمال کرتے ہوئے پہنچا جاتا ہے۔	01:41

اس سے کوئی فرق نہیں پڑتا کہ حروف کی اصلی سٹریم فائل سے آتی ہے یا فائل میں جاتی ہے، یا کی بورڈ، ونڈو وغیرہ سے آتی ہے۔	01:44
لینکس میں پروسیس کی ہر کھلی فائل، عددی نمبر کے ساتھ منسلک ہوتی ہے۔	01:51
اس عددی ویلیو کو فائل ڈسکرپٹر کے طور پر جانا جاتا ہے۔	01:57
لینکس شیل میں IO سٹریمز کا استعمال کرتے ہیں	02:05
ان میں سے ہر ایک معروف فائل ڈسکرپٹر منسلک ہوتا ہے۔	02:08
stdin اسٹنڈارڈ ان پٹ سٹریم ہے۔	02:12
یہ کمانڈ کو ان پٹ فراہم کرتا ہے۔	02:15
یہاں فائل ڈسکرپٹر صفر 0 ہے۔	02:17
stdout، اسٹنڈارڈ آؤٹ پٹ سٹریم ہے۔	02:19
یہ کمانڈ سے آؤٹ پٹ ظاہر کرتا ہے، اس کے پاس فائل ڈسکرپٹر 1 ہے۔	02:22
stderr اسٹنڈارڈ ایرر سٹریم ہے، یہ ایرر آؤٹ پٹ کمانڈس سے ظاہر کرتا ہے، یہاں فائل ڈسکرپٹر 2 ہے۔	02:26
ان پٹ سٹریمز پروگراموں کو ان پٹ دستیاب کرتا ہے۔	02:36
ڈفالٹ طور پر ٹرمینل کے سٹروکس سے لیتا ہے۔	02:40
آؤٹ پٹ سٹریمز ٹرمینل پر ڈفالٹ طریقے سے ٹیکسٹ حروف پرنٹ کرتا ہے	02:44
ٹرمینل پہلے ایک ASCII ٹائپ رائٹ یا ڈسپلے ٹرمینل تھا	02:47
لیکن اب گرافک ڈیسکٹاپ پر ایک ٹیکسٹ ونڈو ہے۔	02:52
ہم نے دیکھا کہ ڈفالٹ طریقے سے 3 سٹریمز کچھ فائلوں سے جڑی ہوئی ہیں	02:56
لیکن لینکس میں ہم یہ ڈفالٹ طریقہ تبدیل کر سکتے ہیں۔	03:01
ہم ان 3 سٹریمز کو دیگر فائلوں سے جوڑ سکتے ہیں۔	03:04
اس عمل کو ریڈائریشن کہتے ہیں۔	03:07
اب دیکھتے ہیں کہ 3 سٹریمز میں ریڈائریشن کس طرح ہوتا ہے۔	03:09
پہلے دیکھتے ہیں کہ اسٹنڈارڈ ان پٹ کیسے ریڈائریشن ہو رہا ہے	03:14

03:17	ہم < (لیفٹ اینڈ گڈ بریکٹ) آپریٹر کا استعمال کرتے ہوئے فائل سے سٹینڈرڈ ان (stdin) کو ریڈاریکٹ کریں گے۔ دیکھتے ہیں کس طرح۔
03:22	ہم جانتے ہیں کہ ڈبلیوسی (wc) کمانڈ کا استعمال فائل میں حروف، الفاظ اور لائنوں کی تعداد جاننے کے لئے کیا جاتا ہے۔
03:28	ٹرمینل ونڈو پر ڈبلیوسی (wc) ٹائپ کریں۔
03:31	اب اینٹر دبائیں۔
03:32	کیا ہوتا ہے؟ ہم بلنگنگ کر سہ دیکھیں گے، اس کا مطلب ہے کہ ہمیں بورڈ سے اینٹر کرنے کی ضرورت ہے۔
03:37	کچھ متن اینٹر کریں "This tutorial is very important"
03:46	اب اینٹر دبائیں۔
03:48	اب Ctrl اور d کیساتھ میں دبائیں۔
03:52	اب کمانڈ ان لائنوں پر کام کرے گی جنہیں ہم نے درج کیا ہے۔
03:55	کمانڈ ٹرمینل پر ایک آؤٹ پٹ دے گی۔
03:57	اب یہاں ڈبلیوسی (wc) کمانڈ کے بعد کوئی بھی فائل نہیں دی گئی ہے۔
04:01	تویہ اسٹنڈارڈ ان پٹ سٹریم سے ان پٹ لے گا۔
04:04	اب اسٹنڈارڈ ان پٹ سٹریم خود کار طریقے سے کی بورڈ سے جڑ جاتی ہے۔ اس لئے ڈبلیوسی (wc) کی بورڈ سے ان پٹ لے گا۔
04:12	اب اگر ہم "wc space 'left-angled bracket' space test1 dot txt" لکھتے ہیں
04:19	کیا ہوتا ہے کہ ڈبلیوسی ہمیں فائل test1 dot txt میں حروف، الفاظ اور لائنوں کی تعداد بتائے گا۔
04:27	اب "wc space test1 dot txt" ٹائپ کریں
04:34	ہمیں پھر اسی طرح کا نتیجہ ملتا ہے
04:37	لہذا فرق کیا ہے؟
04:39	جب ہم wc space test1 dot txt لکھتے ہیں کمانڈ فائل test1 dot txt کو کھولتی اور اس سے ریڈ (پڑھتی) کرتی ہے۔

لیکن جب ہم "wc space 'left-angled bracket' test1.txt" لکھتے ہیں، ڈبلیوسی تب بھی کسی فائل کو نہیں کھولتا ہے۔	04:46
اس کے بجائے اسٹینڈرڈ ان (stdin) سے ان پٹ لینے لگتا ہے۔	04:53
اب ہم اسٹینڈرڈ ان کو فائل test1 dot txt میں ڈاریکٹ کرتے ہیں۔	04:57
لہذا کمائنڈ ٹیسٹ 1 سے ریڈ (پڑھتی) کرتی ہے	05:01
لیکن اصل میں یہ اس سے بے خبر ہے کہ اسٹینڈرڈ ان (stdin) میں ڈیٹا کہاں سے آرہا ہے۔	05:04
لہذا ہم نے دیکھا کہ اسٹینڈرڈ ان پٹ کو کیسے ریڈاریکٹ کرتے ہیں	05:10
اب دیکھتے ہیں کہ اسٹینڈرڈ آؤٹ پٹ اور اسٹینڈرڈ ایرر کو کیسے ریڈاریکٹ کرتے ہیں۔	05:12
یہاں فائل کے لئے آؤٹ پٹ یا ایرر کو ریڈاریکٹ کرنے کے دو راستے ہیں۔	05:17
فرض کریں کہ این سے مراد فائل ڈسکرپٹر ہے۔ (n>) این 'سنگل رائٹ - ایگلڈ بریکٹ فائل ڈسکرپٹر این سے فائل تک آؤٹ پٹ ریڈاریکٹ کرتا ہے	05:20
آپ کو فائل کو لکھنے کا حق ہونا چاہئے۔	05:29
اگر فائل موجود نہیں ہے، تو بنائی جاتی ہے،	05:32
اگر یہ موجود ہے تو بغیر کسی نوٹس کی اس میں موجود موضوع ختم ہو جاتے ہیں۔	05:35
این 'ڈبل رائٹ - ایگلڈ بریکٹ' بھی فائل کے لئے فائل ڈسکرپٹر این سے آؤٹ پٹ ریڈاریکٹ کرتا ہے۔	05:40
پھر سے آپ کو فائل کو لکھنے کا حق ہونا چاہئے۔	05:47
اگر فائل موجود نہیں ہے، یہ بنائی جاتی ہے،	05:50
اگر یہ موجود ہے تو آؤٹ پٹ موجودہ فائل سے منسلک ہوتا ہے۔	05:52
این 'سنگل رائٹ - ایگلڈ بریکٹ' یا این 'ڈبل رائٹ - ایگلڈ بریکٹ' میں این سے مراد فائل ڈسکرپٹر ہے	05:59
اگر اسے ہٹا دیتے ہیں تو اسٹینڈرڈ آؤٹ پٹ، فائل ڈسکرپٹر 1 سمجھا جاتا ہے۔	06:05
اس لئے، >(رائٹ - ایگلڈ بریکٹ) >1 کے متوازی ہے	06:10
لیکن ایرر سٹریم کو ریڈاریکٹ کرنے کے لئے >(single right angled bracket) 2 یا >>(double right angled bracket) استعمال کرنا پڑے گا۔	06:15
اب اسے عملی طور پر دیکھتے ہیں	06:22

گزشتہ مثال میں ہم نے دیکھا کہ فائل یا اسٹینڈرڈ ان (stdin) میں ڈبلیوسی کمانڈ کا نتیجہ ٹرمینل ونڈو پر دکھایا جاتا ہے۔	06:24
کیا ہوگا اگر ہم اسے ٹرمینل پر ظاہر نہیں کرنا چاہتے؟	06:31
ہم اسے فائل میں سٹور کرنا چاہتے ہیں۔ تاکہ معلومات بعد میں استعمال کی جاسکے۔	06:34
ڈفالٹ طور سے ڈبلیوسی اپنا آؤٹ پٹ اسٹینڈرڈ آؤٹ (stdout) میں لکھتا ہے۔	06:38
ڈفالٹ طور اسٹینڈرڈ آؤٹ (stdout) ٹرمینل ونڈو سے جڑا ہوتا ہے	06:42
اس لئے ہم آؤٹ پٹ کو ٹرمینل میں دیکھتے ہیں	06:45
لیکن اگر ہم اسٹینڈرڈ آؤٹ (stdout) کو فائل میں ریڈاریکٹ کر سکتے تو ڈبلیوسی کمانڈ سے آؤٹ پٹ کو اس فائل میں لکھا جائے گا۔	06:48
مانیں کہ ہم "wc space test1 dot txt 'right-angled bracket' wc_results dot txt" لکھیں گے۔	06:57
اور اینٹر دبائیں۔	07:09
اب دیکھیں کہ کیا یہ واقعی میں ہوا، ہم wc_results (ڈبلیوسی _ رزلٹ) ڈوٹ ٹیکسٹ کنٹینٹس کو c-a-t کیٹ کمانڈ کے ذریعے ظاہر کر سکتے ہیں۔	07:11
جی ہاں یہ ہوا۔	07:23
مانیں کہ ہمارے پاس اسی ڈاریکٹری میں ٹیسٹ 2 دوسری فائل ہے۔	07:24
اب ہم پھر سے ٹیسٹ 2 فائل کے ساتھ کمانڈ execute کرتے ہیں۔ ہم "wc space test2 dot txt 'right-angled bracket' wc_results dot txt" ٹائپ کرتے ہیں	07:30
لہذا ڈبلیوسی انڈر سکور رزلٹ فائل کے کنٹینٹس overwrite ہو جائیں گے۔	07:44
چلئے دیکھتے ہیں۔	07:48
اس کے بجائے اگر ہم "wc space test1.txt 'right-angled bracket' twice wc_results.txt" لکھیں۔	07:56
اب نئے کنٹینٹس wc_results.txt فائل میں پہلے سے موجود موضوع کوری رائٹ نہیں کرے گا بلکہ اس کے ساتھ شامل ہو جائے گا۔	08:07
چلئے اسے بھی دیکھتے ہیں۔	08:15

08:26	اسٹینڈرڈ ایرر رڈاریکٹ کرنا اسی طرح ہوتا ہے۔
08:29	اب فرق صرف اتنا ہے کہ یہاں پر ہمیں <یا> کے نشان سے پہلے standarderr کے فائل ڈسکرپٹر کی تعداد کو بیان کرنے کی ضرورت ہے۔
08:38	جیسے، ہم جانتے ہیں کہ aaa نام کی کوئی فائل موجود نہیں ہے، اب لکھیں ڈبلیو سی خلائی wc space aaa
08:46	شیل ایرر دے گا "No such file or directory" (ایسی کوئی فائل یا ڈائریکٹری نہیں ہے)
08:50	اب مانیں کہ ہم سکرین پر ایرر میسج نہیں چاہتے۔ وہ کسی دوسری فائل میں رڈاریکٹ کر سکتے ہیں۔
08:55	اس کے لئے ہم "wc space aaa space 2 'right-angled bracket' errorlog.txt" کمانڈ دے سکتے ہیں۔
09:06	اب ایرر ٹریٹمنٹ پر نہیں دیکھے گا، بلکہ یہ فائل errorlog.txt میں لکھا جائے گا۔
09:12	ہم یہ "cat space errorlog dot txt" کمانڈ سے دیکھ سکتے ہیں۔
09:22	اب مانیں کہ میں "cat space bbb space 2 'right-angled bracket' errorlog.txt" کمانڈ کے اجرا میں کوئی دیگر غلطی کرتا ہوں
09:34	تو کمانڈ چلانے سے پچھلا ایرر overwrite ہو جائے گا اور نیا ایرر نظر آئے گا۔
09:39	دیکھئے ("cat space errorlog dot txt")
09:46	لیکن کیا کریں، اگر ہم تمام ایرروں کی فہرست چاہتے ہیں؟ اس کے لئے ہم یہ کمانڈ چلائیں گے wc space aaa space 2 'right-angled bracket' twice errorlog dot txt
09:58	ہم اسے کیٹ کمانڈ کا استعمال کرتے ہوئے جانچتے ہیں۔
10:06	ہم نے دیکھا کہ کس طرح اسٹینڈرڈ آؤٹ، اسٹینڈرڈ ان، اسٹینڈرڈ ایرر تینوں سٹریمس علیحدہ سے رڈاریکٹ اور موثر طور استعمال ہوتی ہیں۔ لیکن اس حقیقت کی اصلی طاقت کا اندازہ اس وقت لگا سکتے ہیں جب ہم ان سٹریموں کا استعمال بیک وقت کرتے ہیں۔
10:20	اس عمل کو pipelining کہتے ہیں۔
10:22	پائپس کا استعمال کمانڈ کی ترتیب بنانے کے لئے کرتے ہیں۔
10:25	پائپ کی ترتیب میں ایک کمانڈ کے آؤٹ پٹ کو دوسری کمانڈ کے ان پٹ سے جوڑتا ہے۔

<p>یہ اس طرح لگ رہا ہے command1 vertical bar command2 hyphen option vertical bar command3 hyphen option1 hyphen option2 vertical bar command4</p>	10:30
<p>مانیں کہ ہم موجودہ ڈائریکٹری میں موجود ڈائریکٹریز اور فائلوں کی مکمل تعداد جاننا چاہتے ہیں۔</p>	10:46
<p>ہم کیا کر سکتے ہیں؟ ہم جانتے ہیں کہ "ls space minus  " موجودہ ڈائریکٹری کی تمام فائلوں اور ڈائریکٹریز کی فہرست بنائے گا۔</p>	10:51
<p>ہم ایک فائل 'right-angled bracket' files.txt کے لئے آؤٹ پٹ رڈاریٹ کر سکتے ہیں۔</p>	10:58
<p>"cat space files dot txt" چلائیں۔</p>	11:08
<p>اب ہر لائن، فائل یا ڈائریکٹری کا نام ہے۔</p>	11:14
<p>اس لئے اگر ہم اس فائل میں تمام لائنوں کو گننا چاہتے ہیں، ہم اپنے مقصد کو مکمل کرنے کے لئے files.txt کا استعمال کر سکتے ہیں۔</p>	11:17
<p>اس کو ہم (wc space minus   files dot txt) کمانڈ کا استعمال کرتے ہوئے کر سکتے ہیں۔</p>	11:24
<p>حالانکہ یہ ہمارے مقصد کو پورا کرتا ہے پر کچھ دشواریاں بھی ہیں۔</p>	11:32
<p>پہلی دشواری یہ ہے کہ ہمیں نیچ کی فائل، فائلس ڈاٹ ٹی ایکس ٹی چاہئے۔</p>	11:35
<p>اگر پہلی کمانڈ زیادہ ڈیٹا پیش کرتی ہے تو یہ فضول میں ڈسک میموری کو ضائع یا بھر سکتی ہے</p>	11:40
<p>اگر ہم بہت کمانڈوں کی سیریز بھی بنانا چاہتے ہیں، یہ طریقہ سست ہے۔</p>	11:46
<p>ہم اسے پائپس کا استعمال کرتے ہوئے بہت آسان کر سکتے ہیں ہم ( ls space minus   'vertical bar' wc space minus   ) لکھتے ہیں۔</p>	11:50
<p>اور ہم زیادہ سہولت کے ساتھ وہی نتائج حاصل کر سکتے ہیں۔</p>	12:01
<p>ایل ایس کمانڈ سے آؤٹ پٹ ڈبلیو سی کمانڈ کے لئے ان پٹ کے طور پر جاتا ہے۔</p>	12:06
<p>ہم پائپس کا استعمال کرتے ہوئے کمانڈ کی لمبی ترتیب بھی ضم کر سکتے ہیں۔</p>	12:10
<p>پائپس کا ایک عام استعمال ملٹی پیچ ڈسپلے (کارکردگی) پڑھنے کے لئے کیا جاتا ہے۔</p>	12:15
<p>cd space slash user slash bin ٹائپ کریں،</p>	12:19

12:24	اب ہم بن ڈاریکٹری میں ہیں۔
12:28	اب   minus   چلائیں،
12:31	اب آوٹ پٹ اچھے انداز میں دکھائی نہیں دیتا، لیکن اگر اسے پائپ کے ساتھ more سے جوڑ دیں تو دیکھ سکتے ہیں۔
12:37	فہرست کے ذریعے سکروول کرنے کے لئے ایٹریڈ بائیں۔
12:41	اس سے باہر آنے کے لئے Q دبائیں۔
12:45	یہ کچھ کمانڈ تھے جو ہمیں فائل میں کام کرنے میں مدد کرتے ہیں۔
12:48	اس کے علاوہ بھی کئی کمانڈس ہیں۔
12:50	اس کے علاوہ ہر کمانڈ جو ہم نے دیکھی کے کئی دیگر آپشنس ہیں۔
12:54	man کمانڈ کا استعمال کرتے ہوئے ان کے بارے میں مزید جاننے کے لئے میں آپ کی حوصلہ افزائی کرتا ہوں۔
12:58	کمانڈس کو سیکھنے کا صحیح طریقہ ہے ان کا بار - بار استعمال کرنا۔
13:04	اب ہم اس ٹوریل کے اختتام کی طرف آگئے ہیں۔
13:07	سپوکن ٹوریل ٹاک ٹو اٹیچر پروجیکٹ کا حصہ ہے جسے قومی تعلیم مشن نے آئی سی ٹی کے ذریعے حمایت کی ہے۔
13:15	اس پر مزید معلومات ہماری درج ذیل ویب سائٹ پر دستیاب ہیں۔
13:19	اس ٹوریل کا ترجمہ اور صدا بندی میں نے یعنی وجاہت احمد نے کی ہے۔ ہمارے ساتھ شامل ہونے کے لئے آپ کا بہت شکریہ۔