অপারেটিং সিস্টেমের ভূমিকা

(Introduction to Operating Systems)

প্রফেসর চেস্টার রেবিয়ার

(Prof. Chester Rebeiro)

কম্পিউটার সায়েন্স এন্ড ইঞ্জিনিয়ারিং

(Department of Computer Science and Engineering)

ইন্ডিয়ান ইনস্টিটিউট অফ টেকনোলজি (Indian Institute of Technology),মাদ্রাজ

সপ্তাহ – 01 বক্তৃতা- 04

সিপিইউ শেয়ারিং

নমস্কার, ভূমিকা সম্পর্কিত ভিডিওতে, আমরা দেখেছি যে অপারেটিং সিস্টেমের সবচেয়ে গুরুত্বপূর্ণ অ্যাপ্লিকেশানগুলির একটি হলো রিসোর্স ম্যানেজমেন্ট। মূলতঃ যেহেতু যে সিস্টেমটি অপারেটিং সিস্টেমে চলছে সেগুলি হ'ল সীমিত পরিমাণে হার্ডওয়্যার রিসোর্স যুক্ত, তাই অপারেটিং সিস্টেম এই সম্পদগুলি একটি কার্যকর উপায়ে ব্যবহার করার দায়িত্বে রয়েছে। সিস্টেমে উপস্থিত সমস্ত সম্পদগুলির মধ্যে সম্ভবত সবচেয়ে গুরুত্বপূর্ণ সম্পদ হল CPU। সুতরাং, বেশিরভাগ সিস্টেমে খুব কম সংখ্যক CPU রয়েছে; পূর্ববর্তী সিস্টেমগুলি বিশেষত পূর্বের 1990-এর দশকে এবং ২000-র দশকের শুরুতে একক প্রসেসর ছিল, যদিও এইসব দিনগুলি 4, 8 বা 16 টি CPU- র দ্বারা সজ্জিত। সুতরাং, এটি গুরুত্বপূর্ণ যে অপারেটিং সিস্টেম নিশ্চিত করে যে সিস্টেমে উপস্থিত এই CPUs পর্যাপ্তভাবে ব্যবহার করা হয়। সুতরাং, আমরা দেখব কিভাবে এই CPU গুলো অপারেটিং সিস্টেম দ্বারা পরিচালিত হয়।

(স্লাইডটাইপ পড়ুন: 01:32)

আজকের সিস্টেমগুলিতে, আমাদের বেশ কিছু অ্যাপ্লিকেশন রয়েছে যা ক্রমাগতভাবে চলছে। সুতরাং, আমরা দেখেছি যে আমরা অফিস অ্যাপ্লিকেশন, ইন্টারনেট এক্সপ্লোরার, স্কাইপ অ্যাপ্লিকেশন এবং পাওয়ার পয়েন্ট অ্যাপ্লিকেশনের মতো কিছু থাকতে পারে যা যারা সকলকেই একটিমাত্র CPU ভাগাভাগি করে ব্যবহার করে। সুতরাং, এটি কিভাবে সম্ভব, মূলত কীভাবে অপারেটিং সিস্টেমটি চালানো উচিত বা কোন সিপিইউতে কোন অ্যাপ্লিকেশনটি চালানো উচিত তা নির্ধারণ করে? অপারেটিং সিস্টেমটি করতে পারে এমন একটি খুব সাধারণ

জিনিস হল এক অ্যাপ্লিকেশনকে CPU- এ লাগানো এবং এটি সম্পূর্ণ না হওয়া পর্যন্ত এটি চালানো। উদাহরণস্বরূপ, এমএস ডস অপারেটিং সিস্টেমের মতো একটি অতিপ্রয়োজনীয় অপারেটিং সিস্টেমটি একটি অ্যাপ্লিকেশন শুরু করবে এবং এটি সম্পূর্ণ না হওয়া পর্যন্ত এটি CPU এ এক্সিকিউট করবে। যখন এই অ্যাপ্লিকেশনটি সমাপ্ত হয় তখন অ্যাপ্লিকেশন 2 এর পরের অ্যাপ্লিকেশনটি সিপিইউ পর্যন্ত কার্যকর হবে যতক্ষন পর্যন্ত এটি সমাপ্ত না হয়, তারপর অ্যাপ্লিকেশন 3 চালু হবে এবং এটিকে সিপিইউ পর্যন্ত চালু করতে হবে এবং তারপর অ্যাপ্লিকেশন 4 চালু হবে। যদিও, এটি CPU- র সময় পরিচালনার একটি খুব সহজ উপায়, এটি খুব কার্যকর নয় এবং আমরা এর কারণ দেখব।

(স্লাইডটাইপ পড়ুন: 03:10)

তাই আসুন আমরা বলতে পারি যে অ্যাপ্লিকেশন 1 সিপিইউতে চালানো হচ্ছে এবং কিছুক্ষণ পরে এটি ব্যবহারকারী দ্বারা করা একটি নির্দিষ্ট অপারেশনের জন্য অপেক্ষা করে। উদাহরণস্বরূপ, অ্যাপ্লিকেশন 1 একটি স্ক্যানের মত একটি ইভেন্টের জন্য অপেক্ষা করছে মূলত এটি ব্যবহারকারীর কীবোর্ডের কিছু ইনপুট করার জন্য অপেক্ষা করছে। তাই, এই সময়ের মধ্যে ব্যবহারকারী আসলে একটি কী না চাপা পর্যন্ত, CPU কিছুই করে না শুধু অলসভাবে অপেক্ষা করছে। তাই মূলত এটি সময় নষ্ট হচ্ছে, তাই এটি হল CPU- র অলস চক্রের সময়। শুধুমাত্র যখন ইভেন্টটি পাওয়া যায়, তখনই অ্যাপ্লিকেশন 1 কে চালানো যাবে। এইভাবে, আমরা দেখতে পাচ্ছি যে এটি খুবই সহজ যে অ্যাপ্লিকেশন 1 চালু হবে অন্যটি সম্পূর্ণ করার পরে, তবে এটি CPU- র সময় পর্যাপ্ত বা কার্যকরীভাবে ব্যবহার করে না, কারণ প্রতিটি সময় অ্যাপ্লিকেশনের জন্য একটি ইভেন্টের প্রয়োজন হয়, CPU প্রকৃতপক্ষে সেই ইভেন্ট না ঘটা পর্যন্ত নিষ্ক্রিয় থাকবে।এবং আমরা দেখি CPU- এর ব্যবহার সম্পর্কে একটি কম কর্মক্ষমতা কারণ হবে।

(স্লাইডটাইপ পড়ুন: 04:46)

সুতরাং একটি ভাল উপায় হল মাল্টিপ্রোগ্রামিং হিসাবে পরিচিত। সুতরাং, একটি অপারেটিং সিস্টেমের মধ্যে যে মাল্টিপ্রোগ্রামিং সমর্থন করে, অ্যাপ্লিকেশন 1 এটি একটি বহিরাগত ইভেন্ট প্রয়োজন পর্যন্ত এক্সিকিউশন চালিয়ে যাবে। সুতরাং, scanf এর মত একটি ফাংশন অ্যাপ্লিকেশন 1 এ সঞ্চালিত হবে তখন অ্যাপ্লিকেশনটি অবরোধ করা হবে। মূলত এটি প্রসেসর থেকে প্রিএম্পটেড(preempted) হবে এবং এই ক্ষেত্রে অ্যাপ্লিকেশন 2 সিপিইউতে চালানো হবে । তাই কিছুক্ষণের পরে, যখন ব্যবহারকারী কীবোর্ডের মাধ্যমে কোন কিছু ইনপুট করে, তখন এই ইভেন্টটি প্রাপ্ত হয় এবং এই অ্যাপ্লিকেশন 1 এর ফলে একটি সারিতে রাখা হয়। পরে একটি সময়ে, অ্যাপ্লিকেশন 1 কে আবার CPU তে দেওয়া হবে এবং এটি বন্ধ করা হয়েছে যেখানে সেখান থেকে চালানো হবে। মূলত এটি scanf প্রক্রিয়াকরণের সময় বন্ধ ছিল, এবং এটি বন্ধ ছিল যেখানে সেখান থেকে আবার চালিয়ে যেতে হবে। মূলত এখন এটি ব্যবহারকারী কীবোর্ড থেকে ইনপুট গুলি পেয়েছে এবং এটি এই সেই থেকে পরবর্তীতে চালিয়ে যাবে। অতএব,

আমরা দেখেছি যে আমরা সিপিইউটিকে কর্মহীন হতে প্রতিরোধ করছি। মূলত CPU- র উপর অন্য একটি অ্যাপ্লিকেশন চালানোর মাধ্যমে, আমরা CPU- র দ্বারা অর্থহীন ভাবে চলা প্রতিরোধ করা হয় এবং সেইজন্য, কর্মক্ষমতা বৃদ্ধি পায়। তবে, এক্ষেত্রেও একটি সমস্যা আছে।

(স্লাইডটাইপ পড়ুন: 06:41)

এখন, এই বিশেষ পরিস্থিতিটি বিবেচনা করুন যেখানে অ্যাপ্লিকেশন 1 কখনো কখনো রান করে এবং তারপর ব্লক হয়ে যায়। যাইহোক, অ্যাপ্লিকেশন 2 তার কোড এর মধ্যে এমন কিছু আছে; মূলত এটি একটি অসীম লুপ। সুতরাং, 1 যখন অসীমভাবে কাজ চালিয়ে যাবে। ফলে, এই অ্যাপ্লিকেশন 2 কখনও বন্ধ হবে না, এটা এক্সিকিউট হবে এবং CPU- এর সম্মুখের রাখা হবে ।সুতরাং, এই অ্যাপ্লিকেশন 3, অ্যাপ্লিকেশন 4 এবং সেইসাথে অ্যাপ্লিকেশন 1, যেমন একবার যখন এটি প্রাপ্ত করা হয় ইভেন্ট হিসাবে অন্যান্য অ্যাপ্লিকেশনটি চালানোর জন্য সক্ষম হবে না। এই নির্দিষ্ট জিনিসটি বন্ধ করার একমাত্র উপায় জোরপূর্বক অ্যাপ্লিকেশন 2 কে বন্ধ করা হবে বা সিস্টেম পুনরায় বুট করা হবে। আগের অপারেটিং সিস্টেম যেমন এমএস ডস এ, এই ব্যাপারটি বেশ ঘন ঘন ঘটবে। এটির ফলস্বরূপ, এটি সর্বদা প্রয়োজন যে CPU টি পুনঃসূচনা করা উচিত কারণ একটি অ্যাপ্লিকেশন সমগ্র সিস্টেমটিকে অকেজো করে দেবে।

(স্লাইডটাইপ পড়ুন: 07:55)

আরও সাম্প্রতিক অপারেটিং সিস্টেম সমর্থন করে মাল্টিটাস্কিং বা টাইম শেয়ারিং। মূলত, এই ক্ষেত্রে, সিপিইউ সময় স্লাইসের মধ্যে বিভক্ত হয় বা তাই প্রতিটি টুকরা একটি সময় স্লাইস বা একটি সময় কোয়ান্টা হিসাবে পরিচিত হয়। প্রতিটি সময় টুকররোর মধ্যে, একটি প্রসেস চালানো হবে; প্রসেসটি এক্সিকিউট হতে থাকবে যতক্ষণ না পর্যন্ত এই দুই ঘটনার একটি ঘটে। প্রথমত, এর সময় স্লাইস সম্পূর্ণ হওয়ার আগে পর্যন্ত এটি চালানো হবে; এই ক্ষেত্রে আরেকটি প্রসেস CPU এ দেওয়া হয়। উদাহরণস্বরূপ, এখানে প্রসেস 1 সঞ্চালিত হয় যতক্ষণ পর্যন্ত এর সময় স্লাইস সম্পূর্ণ না হয়, এবং তারপর প্রসেস 2 CPU তে দেওয়া হয় এবং প্রসেস 2 চলতে থাকবে।তারপর যখন প্রসেস 2 তার সময় স্লাইড সমাপ্ত করে, প্রসেস 3 CPU তে দেওয়া হবে এবং এক্সিকিউশন চালিয়ে যাবে। এখন, এই পদ্ধতিতে সব প্রসেস CPU ভাগ করছে। কিছুক্ষণের পরে প্রসেসগুলি আবার এক্সিকিউশন চালিয়ে যেতে পারে যেখানে এটি বন্ধ হয়ে গিয়েছিল। উদাহরণস্বরূপ প্রসেস 1 তার সময় স্লাইড পর্যন্ত এক্সিকিউশন চালিয়ে যাবে এবং অপারেটিং সিস্টেম কিছু সিদ্ধান্তের উপর ভিত্তি করে আবার প্রসেস 1 কে পরে CPU তে দেওয়া হয়। এই সময়ে প্রসেস 1 যেখানে বন্ধ ছিল সেখান থেকে আবার চালু হবে। ব্যবহারকারীর পক্ষে, ম্যালিটাস্কিংয়ের ফলে এই বিলম্বটি অন্য প্রসেসে এক্সিকিউট হয় বা অন্য কোনও ভাবে বলা যায়, একক প্রসেসে বিলম্বিত বা মধ্যবর্তী এক্সিকিউশন খুব কষ্ট করে খুজতে হয় কারণ সময় স্লাইস খুব ছোট। আরেকটি উপায় যা একটি প্রসেস এক্সিকিউশন বন্ধ করে দেয় যখন এটি একটি ইভেন্টের জন্য অপেক্ষা করার প্রয়োজন হয় যেমন আমরা পূর্ববর্তী

স্লাইডে দেখেছি বা এটি বন্ধ হয়ে যায়। অন্য কোনও পদ্ধতিতে এটি CPU- র নতুন প্রসেস এবং নতুন প্রসেসটি চালানো হবে। সুতরাং, মাল্টিটাস্কিংয়ের সুবিধা হল যে ব্যবহারকারীর দৃষ্টিকোণ থেকে, এটি বলা যায় যে সমস্ত অ্যাপ্লিকেশন একসঙ্গে চলছে, এখানে কোন স্টারভেশান(starvation) নেই। এছাড়াও একটি সিস্টেমের সমস্ত দৃষ্টিকোণ থেকে সামগ্রিক কর্মক্ষমতা উন্নত হয়।

(স্লাইডসময় পড়ুন: 10:52)

সুতরাং, CPU- ছিল অপারেটিং সিস্টেম দ্বারা পরিচালিত একটি রিসোর্সের উদাহরণ। সুতরাং, মূলত অপারেটিং সিস্টেমটি সিদ্ধান্ত নিয়েছে যে কোন প্রসেসকে CPU এ এক্সিকিউট করার সুযোগ দেওয়া হবে। তাই, সিপিইউ সহ, সিস্টেমে বেশ কিছু অন্যান্য রিসোর্স থাকবে; এবং অপারেটিং সিস্টেম বিভিন্ন ব্যবহারকারীদের মধ্যে এই রিসোর্সগুলিকে ভাগ করতে হবে যে বিভিন্ন অ্যাপ্লিকেশনের মধ্যে। উদাহরণস্বরূপ, সিস্টেমের সাথে সংযুক্ত প্রিন্টারগুলি, কীবোর্ডগুলি, RAM, ডিস্কগুলি, ডিস্কের ফাইলগুলির পাশাপাশি নেটওয়ার্কগুলিও। এই সব সিস্টেমের রিসোর্সগুলো এবং অপারেটিং সিস্টেম এই রিসোর্সগুলোকে পরিচালনা করা উচিত এবং এটা নিশ্চিত করা যে সমস্ত অ্যাপ্লিকেশন এই বিভিন্ন রিসোর্সগুলোকে ন্যায্য ভাবে ব্যবহার করে।

(স্লাইডটাইপ পড়ুন: 11:54)

সুতরাং, মাল্টিপ্রসেসরের ক্ষেত্রে কি হবে? মূলত, যেখানে সিস্টেমের একটি প্রসেসরের চেয়েও বেশি আছে, এরূপ একটি সিস্টেম এরকম কিছু হবে। মূলত, কিছু চিপ বা CPU চিপ হতে হবে এবং প্রতিটি চিপের বিভিন্ন কোর হবে। উদাহরণস্বরূপ, এই ক্ষেত্রে দুটি চিপ আছে, CPU চিপ; এবং প্রতিটি CPU- র দুটি কোর আছে; এবং আরও সম্ভব যে প্রতিটি কোর একাধিক থ্রেড চালায়। সুতরাং, এটি একটি কৌশল দ্বারা অর্জিত হয় যাকে বলে সিমেট্রিক্যাল মাল্টিথ্রেডিং(Symmetrical multithreading) হিসাবে পরিচিত। ইন্টেলের নামকরণে এটি হাইপার থ্রেডিং নামে পরিচিত। সুতরাং এই বিশেষ ব্যবস্থায়, দুটি চিপ আছে, প্রতিটি চিপে মোট দুটি কোর থাকে এবং প্রতিটি কোরে মোট দুটি সিমেট্রিক্যাল থ্রেড থাকে, CPU- কে এই সব রিসোর্স পরিচালনা করতে হয়।

(স্লাইডটাইপ পড়ুন: 13:07)

তাই মূলত এই ক্ষেত্রে, CPU- কে সিস্টেমের সমস্ত কম্পিউটিং পরিবেশ পর্যাপ্তরূপে ব্যবহৃত হয় তা নিশ্চিত করতে হবে। তাই এই ক্ষেত্রে, আমরা সমতুল্যতা হিসাবে কিছু অর্জন করব যে ধরুন আমরা এখানে ২ টি প্রসেসর আছে এবং আসুন আমরা অনুমান করি যে প্রতিটি প্রসেসরের মধ্যে একটি কোর রয়েছে। সুতরাং, এর মানে হল যে অপারেটিং সিস্টেমটি দুটি প্রসেস বা দুইটি অ্যাপ্লিকেশনকে একযোগে প্রসেসরের উপর চালাতে পারে। সুতরাং, একটি অ্যাপ্লিকেশন একটি প্রসেসরে চালানো হবে, অন্য অ্যাপ্লিকেশনটি দ্বিতীয় প্রসেসরে চালানো হবে। সেইসঙ্গে এই প্রসেসরের প্রতিটিতে সময় স্লাইসিং করা সম্ভব হয় অর্থাৎ প্রতিটি প্রসেসরের সময়কে কোয়ান্টায় বা টাইম স্লাইসে

বিভক্ত করা যায়, যেমন আমরা আগে দেখেছি এবং বিভিন্ন অ্যাপ্লিকেশানে ভাগ করেছি। সুতরাং এই ক্ষেত্রে, উদাহরণস্বরূপ, প্রতিটি প্রসেসরের একটি সময় স্লাইস আছে যা নিয়মিতভাবে সম্পন্ন হয়; এবং প্রতিটি নতুন সময় স্লাইসের শুরুতে, প্রসেসরের উপর চলমান অপারেটিং সিস্টেমটি নিশ্চিত করবে যে এই CPU টি চালানোর জন্য অ্যাপ্লিকেশনটি নির্ধারিত হবে।একইভাবে, দ্বিতীয় প্রসেসরের জন্য, সময় স্লাইসও রয়েছে এবং একটি সময় শেষ হওয়ার সাথে সাথে অ্যাপ্লিকেশনটি চালানোর প্রক্রিয়াটি প্রসেসর থেকে প্রত্যাহার করা হবে এবং OS অন্য অ্যাপ্লিকেশনটিকে চালানোর জন্য নির্বাচন করবে। সুতরাং, অপারেটিং সিস্টেমটি এমনভাবে ডিজাইন করা উচিত যাতে অ্যাপ্লিকেশনগুলি একই সময়ে উভয় প্রসেসরের একযোগে এক্সিকিউট হবে না।

(স্লাইডটাইপ পড়ুন: 15:13)

এখন একটি মাল্টিটাস্কিং(multitasking)পরিবেশের সাথে একটি বিশাল সমস্যা, যেখানে প্রতিটি প্রসেসরের সময় কোয়ান্টা বা সময় স্লাইসগুলিতে বিভক্ত হয় যা বিভিন্ন প্রসেসর বা বিভিন্ন অ্যাপ্লিকেশন একযোগে চালানোর অনুমতি দেয়। এবং যখন সিস্টেমে একাধিক প্রসেসর উপস্থিত থাকে, তখন সমান্তরালে দুই বা ততোধিক প্রসেসর বা অ্যাপ্লিকেশন চালানো যায়। সুতরাং, এখানে উত্থাপিত সমস্যাটি যখন দুটি প্রসেসর বা দুটি অ্যাপ্লিকেশন একযোগে কিছু সম্পদ অ্যাক্সেস করার জন্য অনুরোধ করা হয়। উদাহরণস্বরূপ, আসুন আমরা বলি এই রিসোর্সটি একটি বিশেষ ফাইল বা এটি এমন একটি ডিভাইস যা প্রিন্টারের মতও হতে পারে, এবং আমাদের কাছে দুটি অ্যাপ্লিকেশন রয়েছে - অ্যাপ্লিকেশন 2 এবং অ্যাপ্লিকেশন 5 এই রিসোর্সে কিছু লিখতে বা প্রিন্ট করতে চায়। সুতরাং, অ্যাপ্লিকেশন 2 এবং অ্যাপ্লিকেশন 5 উভয় এই নির্দিষ্ট রিসোর্স ব্যবহার করতে চান। সুতরাং, এটি একটি রেস কন্ডিশন হিসাবে পরিচিত হয়; এবং এর ফলে একটি বিশাল সমস্যা তৈরী হয় যখন আমরা অপারেটিং সিস্টেম পড়ব বা যখন আমরা অপারেটিং সিস্টেমের আরও গভীরে প্রবেশ করব। সুতরাং মূলত, এই ধরনের একটি সমস্যা এড়ানোর জন্য, অপারেটিং সিস্টেম দুটি অ্যাপ্লিকেশন মধ্যে সমন্বয় করা প্রয়োজন। এটা নিশ্চিত করা উচিত যে যখন একটি নির্দিষ্ট প্রসেস একটি অনুরোধ করেছে যেমন হার্ড ডিস্কে সংরক্ষিত ফাইলের বা প্রিন্টারে মতো, অন্য একটি অ্যাপ্লিকেশন 5 এই নির্দিষ্ট সম্পদটি লিখতে বা অ্যাক্সেস করার অনুমতি পাবে না। শুধুমাত্র যখন অ্যাপ্লিকেশন 2 তার রিসোর্স ব্যবহার সম্পূর্ণ করবে, তখনই অ্যাপ্লিকেশন 5 কে সেই সম্পদ ব্যবহার করার অনুমতি দেওয়া হবে; অন্যভাবে বললে অপারেটিং সিস্টেম এই রিসোর্সের একটি ধারাবাহিক এক্সেস নিশ্চিত করবে। এটি নিশ্চিত করবে যে একাধিক অ্যাপ্লিকেশন একটি বিশেষ তাত্ক্ষণিক সময়ে এই রিসোর্স ব্যবহার করে না।

(স্লাইডটাইপ পড়ুন: 17:46)

মূলত, অপারেটিং সিস্টেমগুলি এই সমস্যাগুলিকে সিঙ্ক্রোনাইজেশন নামে পরিচিত একটি কৌশল ব্যবহার করে সমাধান করে। এখন, সিঙ্ক্রোনাইজেশন আপনি মনে করতে পারেন একটি লক ধরনের, যা একটি রিসোর্সের সঙ্গে

যুক্ত।সুতরাং, যখন একটি অ্যাপ্লিকেশন যে রিসৌর্স অ্যাক্সেস করতে চায় তখন এটি প্রথমে লক পেতে হবে; মূলত এটি প্রথম লক অর্জন করা উচিত।উদাহরণস্বরূপ, আমাদের অ্যাপ্লিকেশন X আছে যা রিসৌর্স ব্যবহার করতে চায়; এই ক্ষেত্রে একটি অ্যাপ্লিকেশন X এর রিসৌর্সটিকে বন্ধ করা উচিত। সুতরাং, রিসৌর্স লক করা মানে এই পদ্ধতিতে অ্যাপ্লিকেশন Y বা 5 হিসাবে অন্য কোন অ্যাপ্লিকেশন এই রিসৌর্স ব্যবহার করতে সক্ষম হবে না। এখন অ্যাপ্লিকেশন 2 এর রিসৌর্সের ব্যবহার সম্পন্ন হলে, এটি রিসৌর্সটিকে আনলক করবে। এবং সেই সময়, যদি আমাদের দ্বিতীয় অ্যাপ্লিকেশন থাকে যা রিসৌর্সটির জন্য অনুরোধ করেছে তবে দ্বিতীয় অ্যাপ্লিকেশনটিকে অপেক্ষা করতে হবে। সুতরাং, যখন অ্যাপ্লিকেশন 2 সম্পূর্ণ হয়, এটি রিসৌর্স আনলক করবে; এবং এটি অ্যাপ্লিকেশন 5 এর কাছে একটি সংকেত যে সে তারপর এই রিসৌর্স ব্যবহার করতে পারবে। যাতে অ্যাপ্লিকেশন 5 সম্পদ ব্যবহার করে, এটি প্রথমে রিসৌর্সটি লক করবে যাতে অন্য কোনও অ্যাপ্লিকেশন রিসৌর্সটি ব্যবহার করতে না পারে তা নিশ্চিত করে; এবং তার ব্যবহারের শেষে, এটি রিসৌর্সকে আনলক করবে। আনলক করার এই পধ্যতিটি অন্য অ্যাপ্লিকেশানগুলিকে এই শেয়ার করা রিসৌর্সকে ব্যবহার করতে অনুমতি দেবে।

(স্লাইডটাইম পড়ুন: 19:২7)

এখন, একাধিক টাস্কিং বা মাল্টি-প্রোগ্রাম ভিত্তিক সিস্টেমে অপারেটিং সিস্টেমে বিশেষ করে একটি সিদ্ধান্ত নেওয়া হয় যে পরবর্তী কোন অ্যাপ্লিকেশানটি চালানো উচিত। সুতরাং, আমরা একটি মাল্টিটাস্কিং(multitasking)পরিবেশে দেখেছি অ্যাপ্লিকেশন 1 তার সময় স্লাইস পর্যন্ত চালানো হবে এবং অপারেটিং সিস্টেম তারপর প্রসেস বা কোন অ্যাপ্লিকেশন পরবর্তী চালানো উচিত তা সিদ্ধান্ত নেয়। সুতরাং, এই সিদ্ধান্ত অপারেটিং সিস্টেমের একটি সত্তা দ্বারা নির্ধারিত হয় যাকে বলা হয় সিডিউলার(Scheduler)। মূলত, সিডিউলারকে এমনভাবে ডিজাইন করা হবে যে এটি ন্যায্য হতে হবে যার মানে হল যে এটি এমনভাবে ডিজাইন হওয়া উচিত যে, প্রতিটি পদ্ধতিতে বা এই পদ্ধতিতে চালানো প্রত্যেকটি অ্যাপ্লিকেশনটি CPU- তে যথাযথ অংশ গ্রহণ করা উচিত। কিছু বিশেষ সিস্টেমে, সিডিউলারকে এইভাবে ডিজাইন করা উচিত যেন অন্যদের উপর কিছু অ্যাপ্লিকেশনকে অগ্রাধিকার দেওয়া হয়। সুতরাং এটা বোঝানো হল যে আমাদের কিছু অ্যাপ্লিকেশন থাকতে পারে যা অন্যান্য অ্যাপ্লিকেশনের তুলনায় অনেক বেশি গুরুত্বপূর্ণ এবং এই অ্যাপ্লিকেশনগুলোকে অগ্রাধিকার দেওয়া প্রয়োজন। অন্য কথায়, এই উচ্চ অগ্রাধিকার অ্যাপ্লিকেশনগুলোকে আরো CPU সময় দেওয়া উচিত পাশাপাশি এটি নিশ্চিত করা উচিত যে এই উচ্চ অগ্রাধিকার অ্যাপ্লিকেশনগুলো CPU- এ চালানোর জন্য খুব দীর্ঘ অপেক্ষা না করা উচিত। উদাহরণস্বরূপ, ধরা যাক আমাদের তিনটি অ্যাপ্লিকেশন আছে - অ্যাপ্লিকেশন 1, 3 এবং 4, যা কম অগ্রাধিকারযুক্ত অ্যাপ্লিকেশন। উদাহরণস্বরূপ, এটি একটি কম্পাইলার এক্সিকিউটিং বা একটি সংখ্যার মৌলিকত্ব নির্ধারন করার মত কোনও বৈজ্ঞানিক অপারেশন হতে পারে। সুতরাং, এটি একটি নিম্ন অগ্রাধিকার অ্যাপ্লিকেশন হিসাবে বিবেচনা করা যেতে পারে। যদিও প্রসেস 2 আমাদের অনুমান করা একটি উচ্চ অগ্রাধিকারযুক্ত

অ্যাপ্লিকেশন, এটি একটি রোবট বলে কিছু প্যারামিটার নিয়ন্ত্রণ করতে পারে বা এটি পরিবেশ সম্পর্কে কিছু তথ্য অর্জন করতে পারে যেমন কোনও রাষ্ট্রের তাপমাত্রা বা আর্দ্রতার মান। এখন, এই অ্যাপ্লিকেশনটি একটি উচ্চতর অগ্রাধিকার হওয়ার কারণে, অপারেটিং সিস্টেমটিকে নিশ্চিত করতে হবে যে এই অ্যাপ্লিকেশন 2টি এক্সিকিউট করার জন্য CPU এর আরো সময় দেওয়া উচিত এবং সেইসাথে এই অ্যাপ্লিকেশনটিকে CPU টাইম পাওয়ার আগে অপেক্ষা করতে হবে না। সুতরাং, প্রসেসরগুলির সংখ্যার ওপর ভিত্তি করে এই সমগ্র সিদ্ধান্তগুলি হয় এবং CPU- র প্রতিটি কোরের সংখ্যা এবং থ্রেডের সংখ্যা এবং প্রসেসরটি কীভাবে এক্সিকিউট করা উচিত, যার মধ্যে সিপিইউ এবং সিডিতে থাকা থ্রেড সিডিউলার দ্বারা নির্ধারিত হয়, যা অপারেটিং সিস্টেম নিয়ন্ত্রণ করে।

(স্লাইডটাইপ পড়ুন: 23:00)

অপারেটিং সিস্টেমের কথা বলার সময় অন্যান্য গুরুত্বপূর্ণ প্রয়োজনীয়তা হল আইসোলেশন(Isolation) প্রয়োজন। মূলত, এই থেকে উদ্ভূত হয় যে আমাদের একাধিক অ্যাপ্লিকেশন আছে যেগুলি একযোগে বা সিস্টেমের মধ্যে স্লাইস (slice) পদ্ধতিতে চালানো হবে। এখন এই অ্যাপ্লিকেশনগুলি বিভিন্ন ব্যবহারকারীদের কাছ থেকে হতে পারে। এছাড়াও কিছু ম্যালিসিয়াস অ্যাপ্লিকেশনগুলি হতে পারে যেমন এই অ্যাপ্লিকেশানটি একটি ভাইরাস বা ট্রোজান হতে পারে যা সিস্টেমের মধ্যে নিজে থেকেই পরিচালিত হয় আবার CPU- এর এক্সিকিউসন ও পরিচালনা করে। অতএব, এক অ্যাপ্লিকেশনকে অন্য অ্যাপ্লিকেশন থেকে রক্ষা করার জন্য আইসোলেশন প্রয়োজন এবং এর ফলে একটি অ্যাপ্লিকেশনের তথ্য অন্য অ্যাপ্লিকেশন থেকে দৃশ্যমান হতে পারে না। এবং আইসোলেশনের জন্য আর যা প্রয়োজন তা অপারেটিং সিস্টেমের কার্নেল সংক্রান্ত। সুতরাং, সিস্টেম এমনভাবে ডিজাইন করা হয় যে সিস্টেমের মধ্যে এক্সিকিউট হয় এমন প্রসেসর বা অ্যাপ্লিকেশন সরাসরি বিভিন্ন রিসোর্স ব্যবহার করে না। উদাহরণস্বরূপ, সিস্টেমে চলমান একটি অ্যাপ্লিকেশনকে একটি যন্ত্র যেমন প্রিন্টারকে সরাসরি অ্যাক্সেস প্রদান করা হবে না। অ্যাপ্লিকেশনটি যদি প্রিন্টার ব্যবহার করতে চায়, তাহলে অপারেটিং সিস্টেমের মাধ্যমে যেতে হবে, এটি একটি সিস্টেম কল করা প্রয়োজন যা প্রিন্টারগুলির সাথে সম্পর্কিত ডিভাইস ড্রাইভার চালানোর জন্য অপারেটিং সিস্টেমকে ঘুরিয়ে দেবে। অ্যাপ্লিকেশন তারপর ডকুমেন্ট বা টেক্সট ফাইল পাঠাতে হবে অপারেটিং সিস্টেমকে যা কিনা প্রিন্ট করতে হবে, তারপর এটি প্রিন্টারে স্থানান্তর হয়।

(স্লাইডসময় পড়ুন: 25:08)

সুতরাং, এটি অর্জন করার জন্য, অধিকাংশ প্রসেসর এবং অধিকাংশ অপারেটিং সিস্টেমে স্ট্রাকচারের মত রিং আছে। মূলত, ইন্টেল প্রসেসরের ইন্টেল প্ল্যাটফর্মের মধ্যে রয়েছে 4 টি রিং - রিং 0, রিং 1, রিং ২, এবং রিং 3। এখন, রিং 0 যেখানে অপারেটিং সিস্টেম চালানো হয়। সুতরাং, এটি হল অপারেটিং সিষ্টেমের সবচেয়ে বেশি সুবিধাভোগী মোড । তাই রিং 0 এ, অপারেটিং সিস্টেম বা কার্নেল যা এখানে এক্সিকিউট হয় বিভিন্ন সংস্থান পরিচালনার মতো বিভিন্ন কিছু করতে পারে, বিভিন্ন ডিভাইসের সাথে সরাসরি যোগাযোগ করতে পারে এবং

কার্য সম্পাদনকারী বিভিন্ন অ্যাপ্লিকেশনের নিয়ন্ত্রণ করতে পারে। সুতরাং আধুনিক অপারেটিং সিস্টেম যেমন লিনাক্স, যা ইন্টেল প্রসেসরের উপর চালিত হয়, তৃতীয় রিং বা রিং 3 ব্যবহারকারী অ্যাপ্লিকেশন দ্বারা ব্যবহৃত হয়। সুতরাং, এটি হল সবচেয়ে কম প্রিভিলেজড রিং এবং এটি নিশ্চিত করে যে এই রিংটিতে চালানো অ্যাপ্লিকেশনগুলি কার্নেলকে অ্যাক্সেস করে না তাই এই রকমের স্বাভাবিক অবস্থায় চলা একটি অ্যাপ্লিকেশন অপারেটিং সিস্টেম সম্পর্কে কিছু দেখতে বা নির্ধারণ করতে সক্ষম হবে না। । সুতরাং, যখনই আপনি আপনার সিস্টেমের উপর একটি অ্যাপ্লিকেশন চালান যেমন আমাদের একটি ওয়েব ব্রাউজার বা অফিস অ্যাপ্লিকেশন বলতে হবে, তাই অ্যাপ্লিকেশন একটি রিং 3 ইউজার অ্যাপ্লিকেশন হিসাবে শুরু করা হয়, এবং যেমন নেটওয়ার্ক বা একটি রিসোর্স ব্যবহার করার জন্য মনিটরে কিছু প্রদর্শন করে, এই অ্যাপ্লিকেশানগুলি সিস্টেম কলের মাধ্যমে কার্নেল আহরণ করতে হয়। অতএব, কার্নেল সমস্ত ব্যবহারকারী অ্যাপ্লিকেশন থেকে বিচ্ছিন্ন হয়। এই ছাড়াও, অপারেটিং সিস্টেম এবং প্রসেসরগুলি এমনভাবে ডিজাইন করা হয় যে ব্যবহারকারীর স্থানগুলিতে চালানো প্রতিটি অ্যাপ্লিকেশন একে অপরের থেকে বিচ্ছিন্ন হয়। এর মানে হল যে যদি আমরা এখানে একটি ওয়েব ব্রাউজার চলছে এবং একটি অফিসের অ্যাপ্লিকেশনের কথা বলে থাকি। প্রসেসর সহ অপারেটিং সিস্টেমের সম্পূর্ণ সিস্টেম নিশ্চিত করবে যে ওয়েব ব্রাউজারের কাছে অন্য অ্যাপ্লিকেশন সম্পর্কে কোন তথ্য নেই অর্থাৎ এটির অফিস অ্যাপ্লিকেশনটি কী করছে তা জানা নেই। এবং অফিস অ্যাপ্লিকেশনটি কীভাবে ওয়েব ব্রাউজার বর্তমানে নির্বাহ করছে তা জানতে পারবে না। উদাহরণস্বরূপ, অফিস অ্যাপ্লিকেশন কখনও জানবে না যে ওয়েব পেজগুলিতে কী ব্রাউজ করা হচ্ছে। সুতরাং, আপনি এটি দেখতে পাবেন যে বিভিন্ন ব্যবহারকারীদের পাশাপাশি আলাদা করাও দরকারী। সুতরাং, আপনার যদি এমন দুটি ভিন্ন ব্যবহারকারী থাকে যারা সিস্টেমটি ভাগ করছে, ব্যবহারকারী 1 নির্দিষ্ট করে না যে ব্যবহারকারী ২ কে কীভাবে কার্যকর করা হচ্ছে যদি না সে নির্দিষ্ট কার্নেল ফাংশন ব্যবহার করে। এবং সেই কার্নেল ফাংশনগুলি ব্যবহার করার একমাত্র উপায় হল সিস্টেম কলগুলির মাধ্যমে অথবা কিছু নির্দিষ্ট ক্ষেত্রে যেমন লিনাক্স, কিছু নির্দিষ্ট কার্যাবলী যা কার্নেল সম্পর্কে প্রদর্শিত হয়। সুতরাং, ব্যবহারকারী তারপর অন্য প্রসেস সম্পর্কে নির্ধারণ করতে কার্নেল এর কার্যকারিতা ব্যবহার করতে সক্ষম হবে।

(স্লাইডটাইপ পড়ুন: ২9:09)

অপারেটিং সিস্টেমের আরেকটি গুরুত্বপূর্ণ দিক হচ্ছে নিরাপত্তার প্রয়োজন। মূলত, নিরাপত্তা বৈশিষ্ট্যগুলি কেবল অপারেটিং সিস্টেমগুলিতে যোগ করা হয় তা নিশ্চিত করার জন্য শুধুমাত্র অনুমোদিত ব্যবহারকারীই এই সিস্টেমটি ব্যবহার করতে সক্ষম। উদাহরণস্বরূপ, যদি আমার কোনও নির্দিষ্ট সার্ভারে কোন ব্যবহারকারীর অ্যাকাউন্ট নেই, তাহলে অপারেটিং সিস্টেমটি নিশ্চিত করবে যে আমি সেই নির্দিষ্ট সার্ভারে লগইন করতে সক্ষম হব না এবং আমি সেই সার্ভারে কোনও অ্যাপ্লিকেশন চালাতে সক্ষম হব না । সুতরাং, সিস্টেমে কীভাবে নিরাপত্তা অর্জন করা হয় তা বিভিন্ন বৈশিষ্ট্য দ্বারা পরিচালিত হয় যা অপারেটিং সিস্টেম সমর্থন করে। উদাহরণস্বরূপ,

একটি কৌশল যা অ্যাক্সেস কন্ট্রোল নামে পরিচিত, যা নির্দিষ্ট ব্যবহারকারী দ্বারা তৈরি করা কিছু ফাইল অন্য ব্যবহারকারীর কাছে দৃশ্যমান নয় তা নিশ্চিত করবে। উদাহরণস্বরূপ, অনুমান করুন আমি গেস্ট অ্যাকাউন্টের মাধ্যমে একটি সিস্টেমে লগইন করি। সুতরাং, অপারেটিং সিস্টেম এই সম্পর্কে জানতে হবে যে আমাকে কী কী ফাইল চালানোর অনুমতি দেওয়া আছে এবং কি কী ফাইল আমি অ্যাক্সেস করতে সক্ষম। সুতরাং, অ্যাক্সেস কন্ট্রোল হিসাবে কিছু জানেন এমন কিছু ব্যবহার করে নিরাপত্তা অর্জন করার একটি উপায়। সুতরাং, অপারেটিং সিস্টেমের অ্যাক্সেস কন্ট্রোল মেকানিজমগুলি বিভিন্ন সংস্থানের অনুমোদন বা পরিচালনা করবে। উদাহরণস্বরূপ, যদি আমি একটি গেস্ট অ্যাকাউন্ট ব্যবহার করে একটি সিস্টেমে লগ ইন করি তাহলে অপারেটিং সিস্টেম এটি সনাক্ত করবে এবং নির্ধারণ করবে যে রিসৌর্সগুলি কী এবং আমি যেগুলি অ্যাক্সেস করতে পারি সেগুলি কি কি। উদাহরণস্বরূপ, ধরে নিই সাধারন অতিথি অ্যাকাউন্টে আমি USB ড্রাইভটি অ্যাক্সেস করতে সক্ষম হব না আর অন্যদিকে আমি হয়তো হার্ডডিস্কে উপস্থিত কিছু ফাইলের শুধুমাত্র পড়ার অ্যাক্সেস পেতে পারি। অপারেটিং সিস্টেমের উপর ভিত্তি করে বা সিস্টেম সম্পর্কিত অন্যান্য ফাইলগুলি পড়লেও লেখা হবে। সুতরাং, আমি এই ফাইলগুলি সব সময়ে দেখতে সক্ষম হবে না। আরেকটি কৌশল, যা নিরাপত্তা অর্জনের জন্য অপারেটিং সিস্টেম দ্বারা ব্যবহৃত হয়, ব্যবহারকারী পাসওয়ার্ড এবং ক্রিপ্টোগ্রাফি। সুতরাং, পাসওয়ার্ড নিশ্চিত করে যে অননুমোদিত ব্যক্তিদের সিস্টেমটি ব্যবহার করার অনুমতি দেয় না। অতএব, আধুনিক ল্যাপটপগুলি বায়োমেট্রিক্স বা ফিংগার প্রিন্ট (finger print)এর মতো কৌশলগুলি ব্যবহার করা হয় সিস্টেমগুলিতে অ্যাক্সেস দেওয়া লোকেদের ফিল্টার করার জন্য ।

(স্লাইডটাইপ পড়ুন: 32:00)

সুতরাং এই বিশেষ স্লাইড দেখায় কিভাবে অ্যাক্সেস নিয়ন্ত্রণকে বাস্তবায়ন করা হয়। সুতরাং, মূলত আমাদের একটি ম্যাট্রিক্স আছে যার প্রতিটি রো(row)একটি ব্যবহারকারীর অনুরূপ। তাই, অ্যান, বব ও কার্ল সিস্টেমের তিনজন ব্যবহারকারী। যদিও কলামগুলি রিসৌর্সগুলিকে নির্দেশ করে যেমন ফাইল 1 যেটি হার্ডডিস্কে থাকে, ফাইল ২, ফাইল 3 এবং প্রোগ্রাম 1। সুতরাং, সিস্টেমের মধ্যে যে ব্যবহারকারীর পাসওয়ার্ডের ভিত্তিতে লগ ইন করা আছে তার উপর নির্ভর করে অপারেটিং সিস্টেমটি প্রতিটি ফাইলের জন্য আলাদা অনুমতি দেওয়া আছে।উদাহরণস্বরূপ, অ্যান ব্যবহারকারী অ্যানের জন্য ফাইল 1 এবং ফাইল 2 হিসাবে পড়া এবং লেখার অনুমোদন রয়েছে এবং প্রোগ্রাম 1 কে এক্সিকিউট করতে পারে। তবে, এনের ফাইল 3 এর জন্য কোনও অনুমতি নেই। অনুরূপভাবে, বব আরেক সিস্টেমের অন্য ব্যবহারকারী শুধুমাত্র ফাইল 1 পড়তে পারেন, কিন্তু তিনি ফাইল 1 এ লিখতে পারেন না, তবে তিনি ফাইল 3 তে লিখতে এবং পড়তে পারেন, এবং অপারেটিং সিস্টেমের অনুমতির উপর ভিত্তি করে বল যায়, ববের প্রোগ্রাম 1 এ কোনও অ্যাক্সেস নেই অর্থাৎ তিনি প্রোগ্রাম 1 কে এক্সিকিউট

করতে পারবেন না। একইভাবে, তৃতীয় ব্যবহারকারী কার্ল ফাইল 1এবং ফাইল 3 তে পড়া বা লেখার কোনও অ্যাক্সেস নেই এবং শুধুমাত্র ফাইল 2 এর অ্যাক্সেস আছে; যাইহোক, কার্ল প্রোগ্রাম 1 এক্সিকিউট করতে পারেন পাশাপাশি প্রোগ্রাম 1 পড়তেও পারেন।

(স্লাইডসময় পড়ুন: 33:46)

এখন আমরা বলতে পারি আমাদের সিস্টেম জুড়ে রযেছে প্রসেসর, বিভিন্ন রিসৌর্স এবং অপারেটিং সিস্টেম যা এই বিভিন্ন রিসৌর্সগুলোকে পরিচালনা করে এবং সিস্টেমগুলি চালানো অ্যাপ্লিকেশনগুলোকেও পরিচালনা করে। সুতরাং, কিভাবে আমরা সিস্টেমের নিরাপত্তা অ্যাক্সেস করতে পারি? এখানে সাধারণভাবে দুটি উপায় আছে।সুতরাং গাণিতিক বিশ্লেষণ বা ম্যানুয়াল বা আধা স্বয়ংক্রিয় যাচাইকরণ কৌশল দ্বারা সম্পন্ন হয়। অতএব, অনেকগুলি সিস্টেম বিশেষ করে যেগুলি সামরিক ও অন্যান্য প্রতিরক্ষা সম্পর্কিত অ্যাপ্লিকেশানগুলির মতো জটিল অ্যাপ্লিকেশানগুলিতে ব্যবহৃত হয়, কঠোর নিরাপত্তা মূল্যায়নের মাধ্যমে এটি নিশ্চিত করে যে অপারেটিং সিস্টেম এবং হার্ডওয়্যার এবং অ্যাপ্লিকেশানগুলি সহ এই সিস্টেমগুলিতে যথেষ্ট পরিমাণ নিরাপত্তা আছে, তাই এগুলোকে জটিল অ্যাপ্লিকেশনের জন্য ব্যবহার করা যেতে পারে। সুতরাং, এই সিস্টেমের মধ্যে একটি টেস্টিং এবং বৈধতা সম্পর্কিত দৃষ্টিভঙ্গিও আছে। ধন্যবাদ।